

# CASAを使った画像解析の紹介

# CASAの基本的な使い方

# CASAの起動

- # casapy <return>

起動メッセージとログウィンドウがあらわれる

```
[screen 0: casapy] — ssh — 84x32
-----> exit()
leaving casapy...
[0:orion]:ttsuka$ casapy           [~/observation/ALMA_SVdata/lecture/images]

=====
The start-up time of CASA may vary
depending on whether the shared libraries
are cached or not.
=====

CASA Version 4.2.1 (r29048)
Compiled on: Tue 2014/04/01 19:49:27 UTC

For help use the following commands:
tasklist          - Task list organized by category
taskhelp          - One line summary of available tasks
help taskname     - Full help for task
toolhelp          - One line summary of available tools
help par.parametername - Full help for parameter name

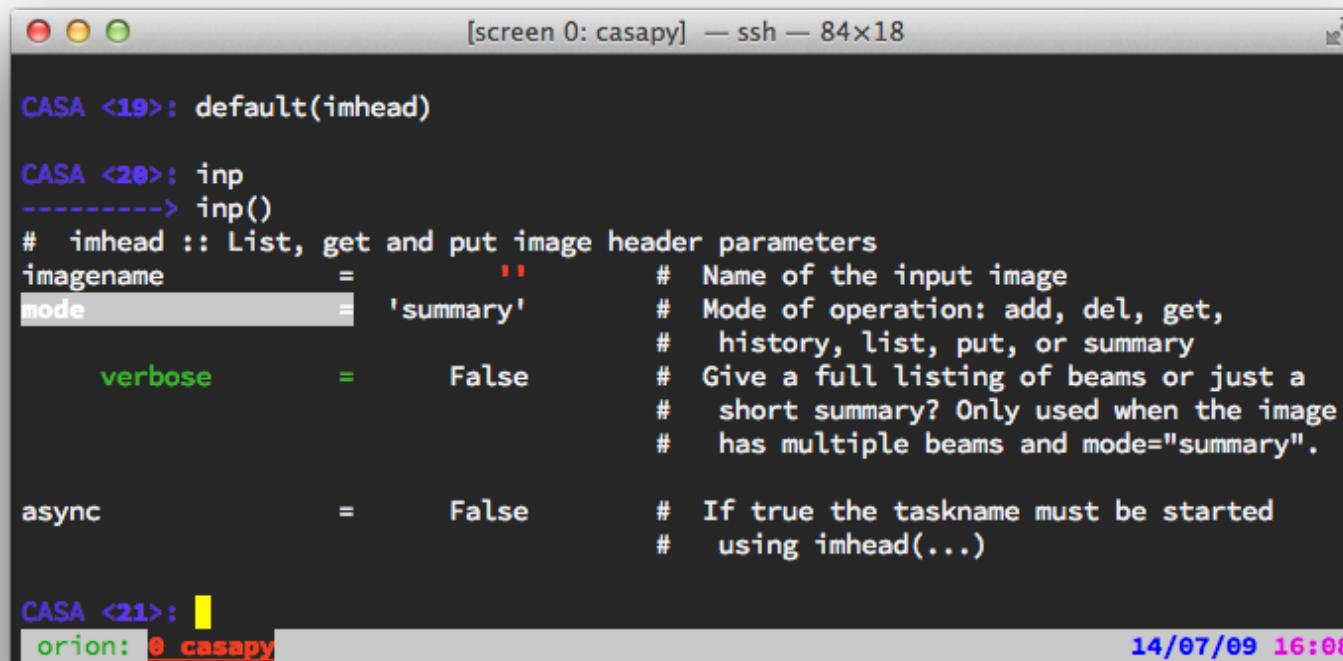
Activating auto-logging. Current session state plus future input saved.
Filename      : ipython-20140709-065942.log
Mode          : backup
Output logging : False
Raw input log  : False
Timestamping   : False
State          : active
*** Loading ATNF ASAP Package...
*** ... ASAP (4.2.0a rev#28807) import complete ***

CASA <2>: [ ]
orion: 0 casapy
```

Log Messages (orion:/home/ttsuka/observation/ALMA_SVdata/lecture/images/casapy-20140709-065937.log)			
Time	Priority	Origin	Message
...	INFO	...sa:::casa	---
...	INFO	...sa:::casa	CASA Version 4.2.1 (release r29047)
...	INFO	...sa:::casa	Tagged on: Thu, 27 Mar 2014

# タスク呼び出し/パラメータ入力/実行

- タスクの呼び出し
  - CASA<>: default([タスク名]) <return> ex) default(imhead) <return>
    - 初期値パラメータで呼び出される
  - CASA<>: tget [タスク名] <return>
    - 前回実行時のパラメータで呼び出される([タスク名].lastファイルが必要)
- タスクパラメータの確認
  - CASA<>: inp <return> or inp([タスク名]) <return>



The screenshot shows a terminal window titled "[screen 0: casapy] — ssh — 84x18". The window displays the following CASA session:

```
CASA <19>: default(imhead)

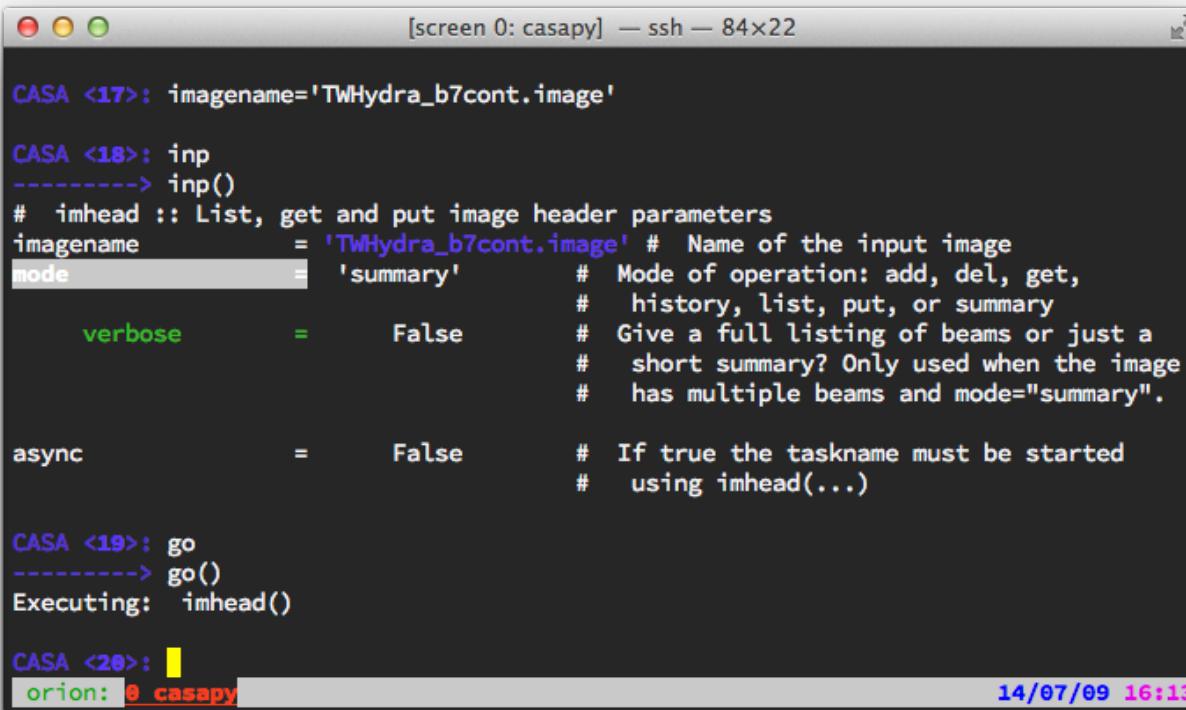
CASA <20>: inp
-----> inp()
# imhead :: List, get and put image header parameters
imagename      =      ""          # Name of the input image
mode           =  'summary'       # Mode of operation: add, del, get,
                                # history, list, put, or summary
verbose         =      False       # Give a full listing of beams or just a
                                # short summary? Only used when the image
                                # has multiple beams and mode="summary".
async          =      False       # If true the taskname must be started
                                # using imhead(...)

CASA <21>: [REDACTED]
orion: 0 casapy
```

The terminal window has a redacted area at the bottom where the user's input was typed. The status bar at the bottom right shows the date and time: "14/07/09 16:08".

# タスク呼び出し/パラメータ入力/実行

- パラメータの入力
  - CASA<>: [パラメータ名]= hogehoge <return>  
ex) imagename='TWHydra\_b7cont.image' <return>
- タスクの実行
  - CASA<>: go <return> / go([タスク名]) <return>
  - スクリプト的なタスクの実行も可能  
ex) imhead(imagename='TWHydra\_b7cont.image') <return>



The screenshot shows a terminal window titled "[screen 0: casapy] — ssh — 84x22". The window displays a CASA script being run. The script starts with setting the image name, followed by defining parameters for the imhead task (mode, verbose, async), and finally executing the task with go(). The status bar at the bottom shows "orion: 0 casapy" and the date/time "14/07/09 16:13".

```
CASA <17>: imagename='TWHydra_b7cont.image'

CASA <18>: inp
-----> inp()
# imhead :: List, get and put image header parameters
imagename      = 'TWHydra_b7cont.image' # Name of the input image
mode           = 'summary'          # Mode of operation: add, del, get,
#                   history, list, put, or summary
verbose        = False             # Give a full listing of beams or just a
#                   short summary? Only used when the image
#                   has multiple beams and mode="summary".

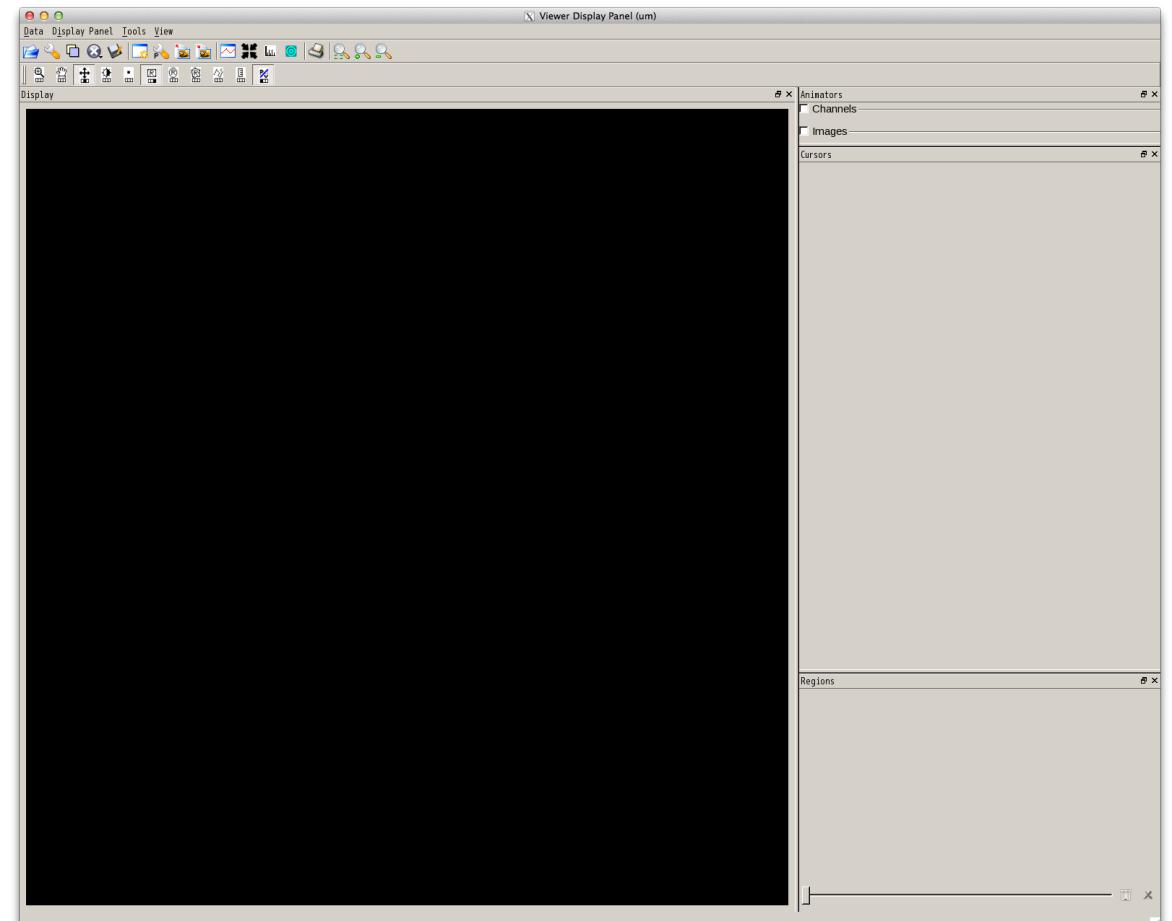
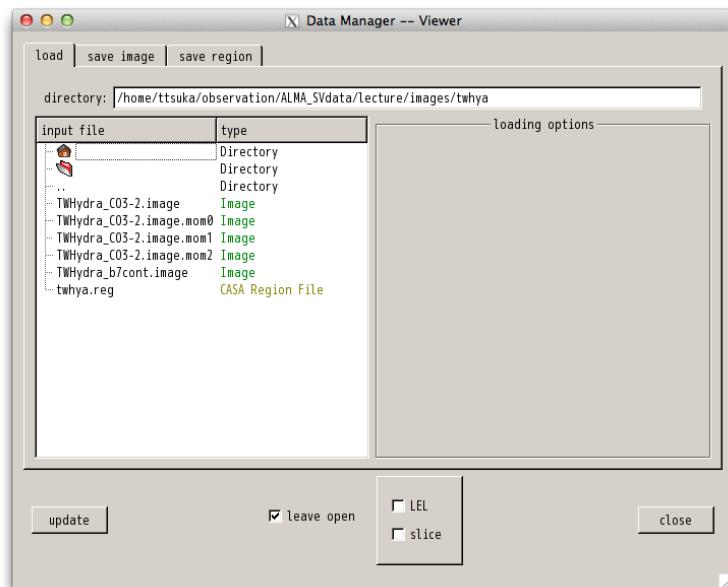
async          = False             # If true the taskname must be started
#                   using imhead(...)

CASA <19>: go
-----> go()
Executing: imhead()

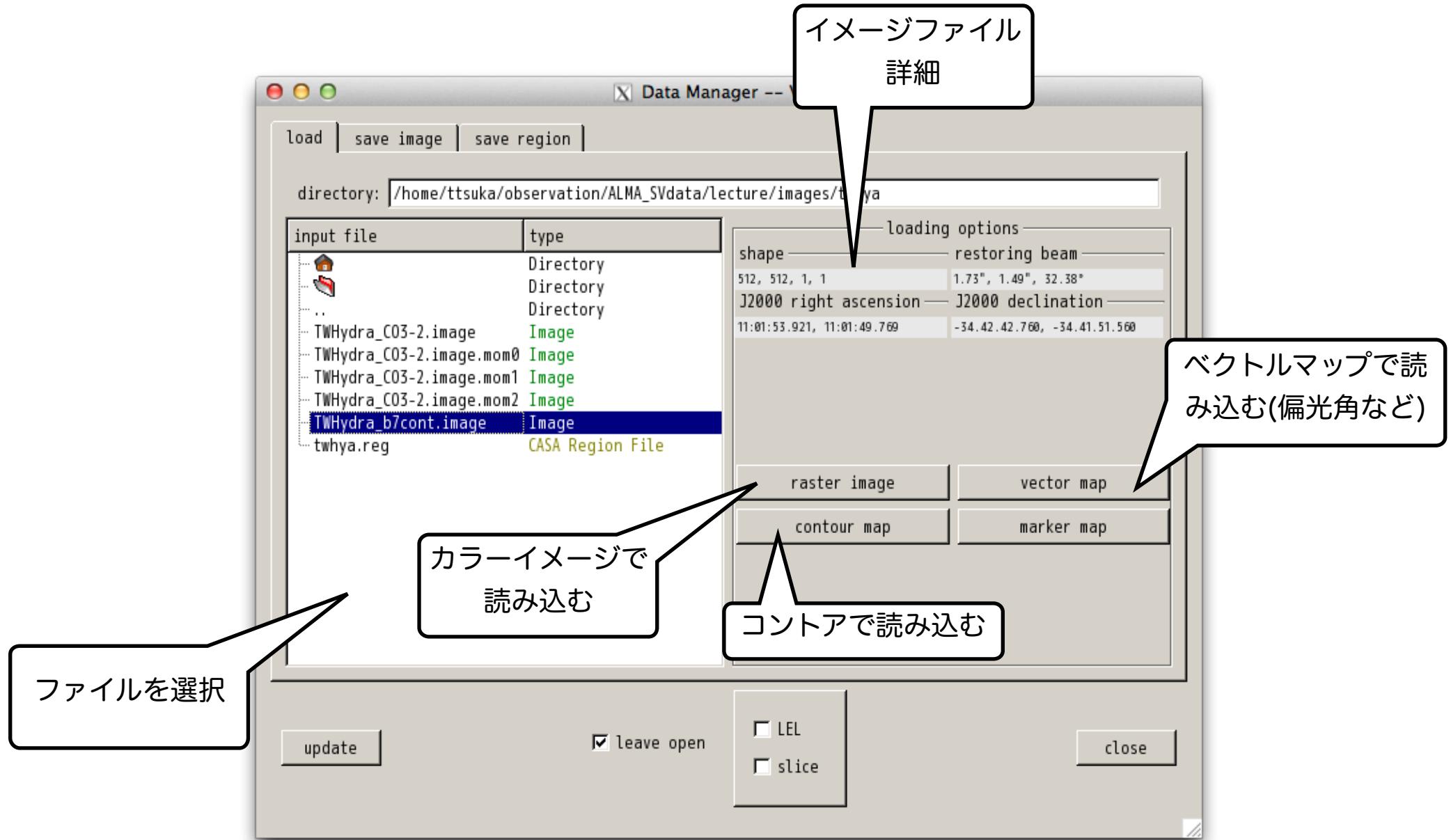
CASA <20>: [redacted]
orion: 0 casapy
```

# CASA viewer

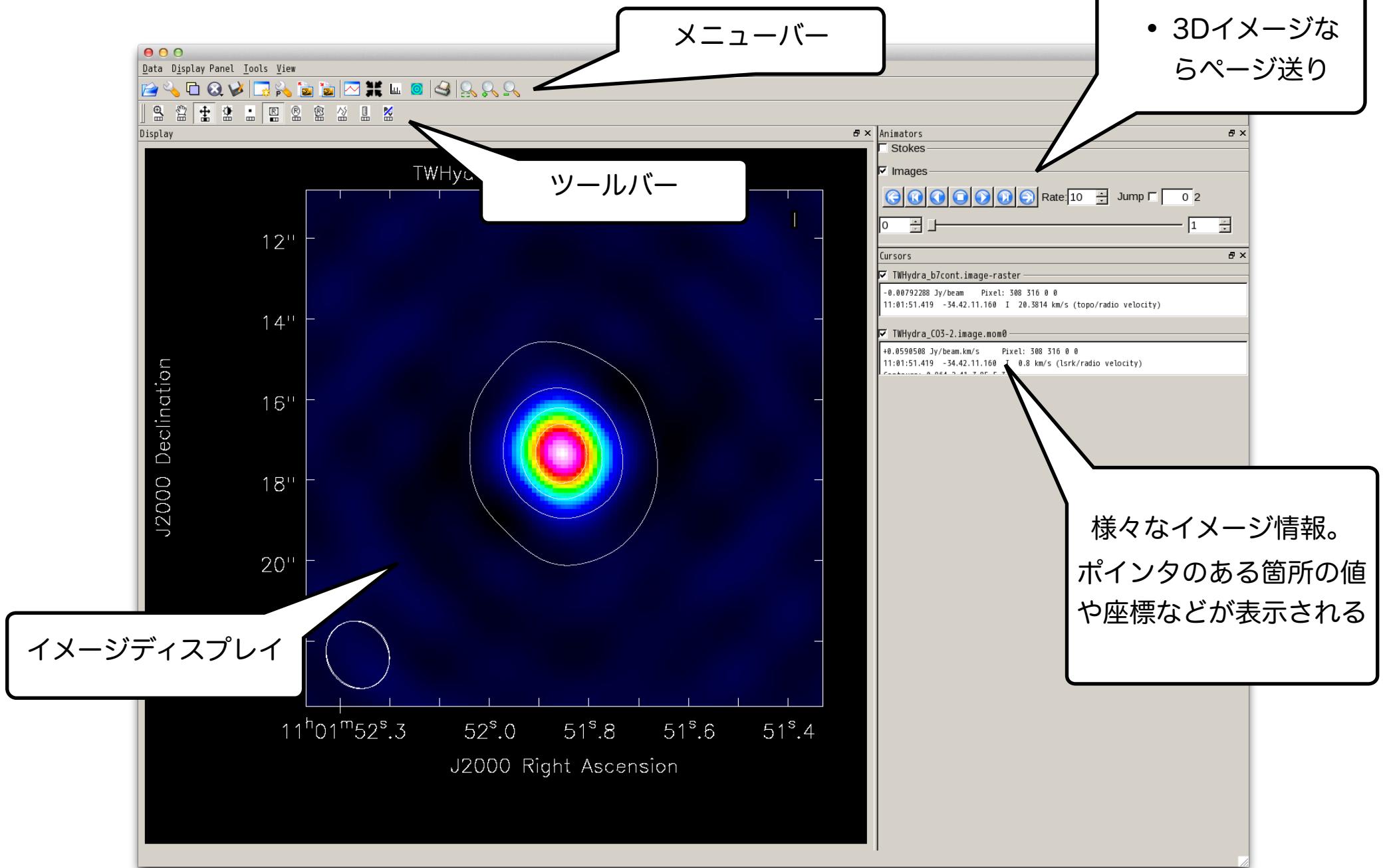
- CASAイメージファイルやFITSイメージを見るツール
- 2通りの起動方法
  - # casapy <return>  
CASA<>: viewer <return>
  - # casaviewer <return>



# CASA viewer: ファイルブラウザ

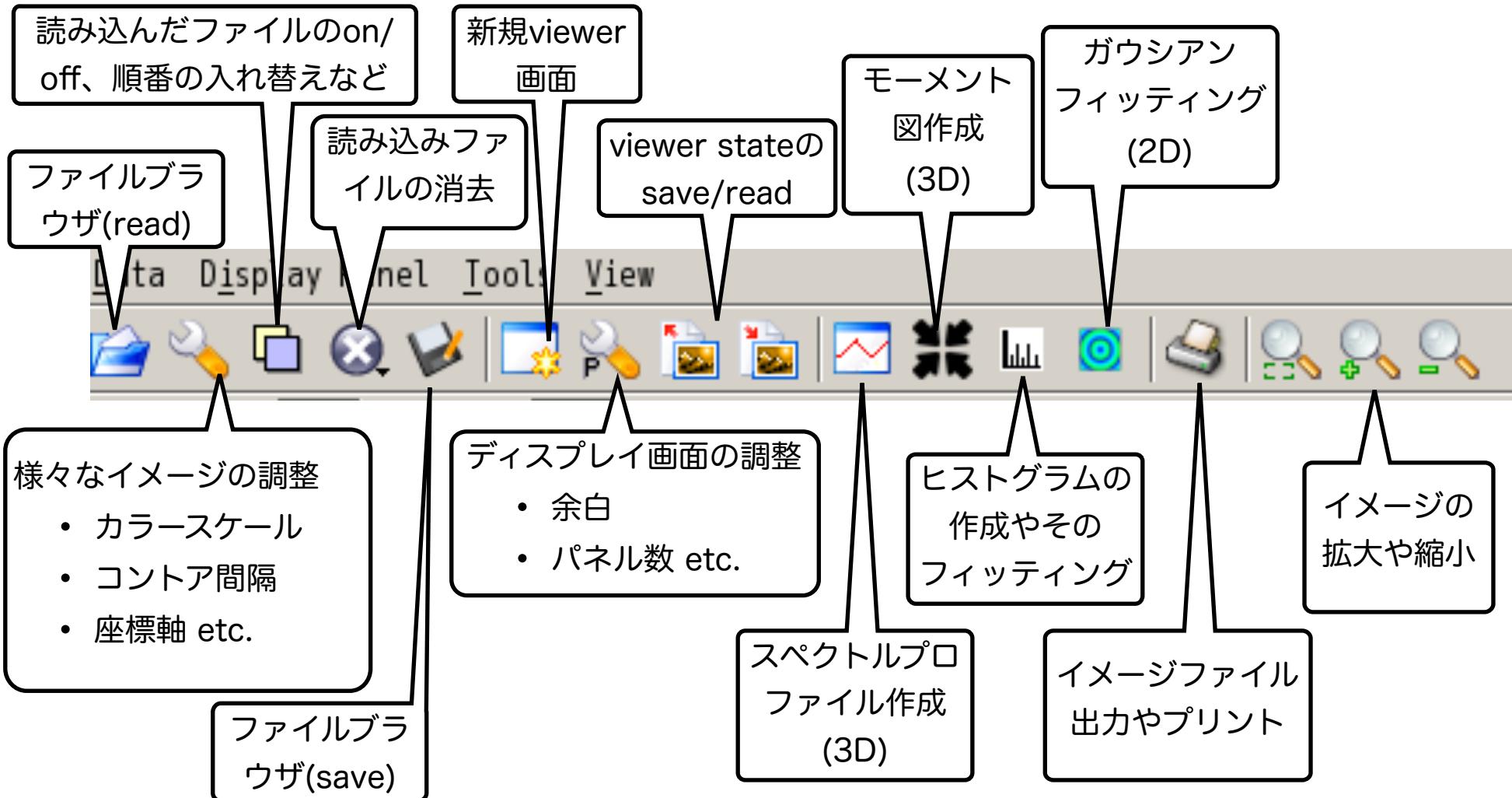


# CASA viewer: メイン画面



# CASA viewer: メニューバー

- ・メニュータブとアイコンは同じ



# CASA viewer: ツールバー

- アイコン下の3つのBOXは、マウスの左/真ん中/右に対応
- 対応するマウスボタンでクリックすると、そのマウスボタンが役割を担うようになる

## イメージの拡大

ドラッグで領域を指定し、その領域範囲内をダブルクリックすると拡大

## カラーレンジ調整

画面内でドラッグするとカラー割り当てが変わる

## ドットを打つ

クリックした場所にドットを打つ。ドット上でESCキーを押すと消える

## 線を引く

連続クリックで線を引いていく。ダブルクリックで指定を終了

## 長さを測る

ドラッグで始点と終点の距離を表示する

## PV図作成

ドラッグで始点と終点を決める。  
Generate PVボタンで作成



## 表示中心の移動

ドラッグで移動距離を指定する

## 明るさ調整

画面内でドラッグすると表示の明るさが変わる

## BOX領域指定

スペクトルやヒストグラムを書く領域の指定。ドラッグで領域を指定する。ESCキーで消える。

## 楕円領域指定

スペクトルやヒストグラムを書く領域の指定。ドラッグで領域を指定する。ESCキーで消える。

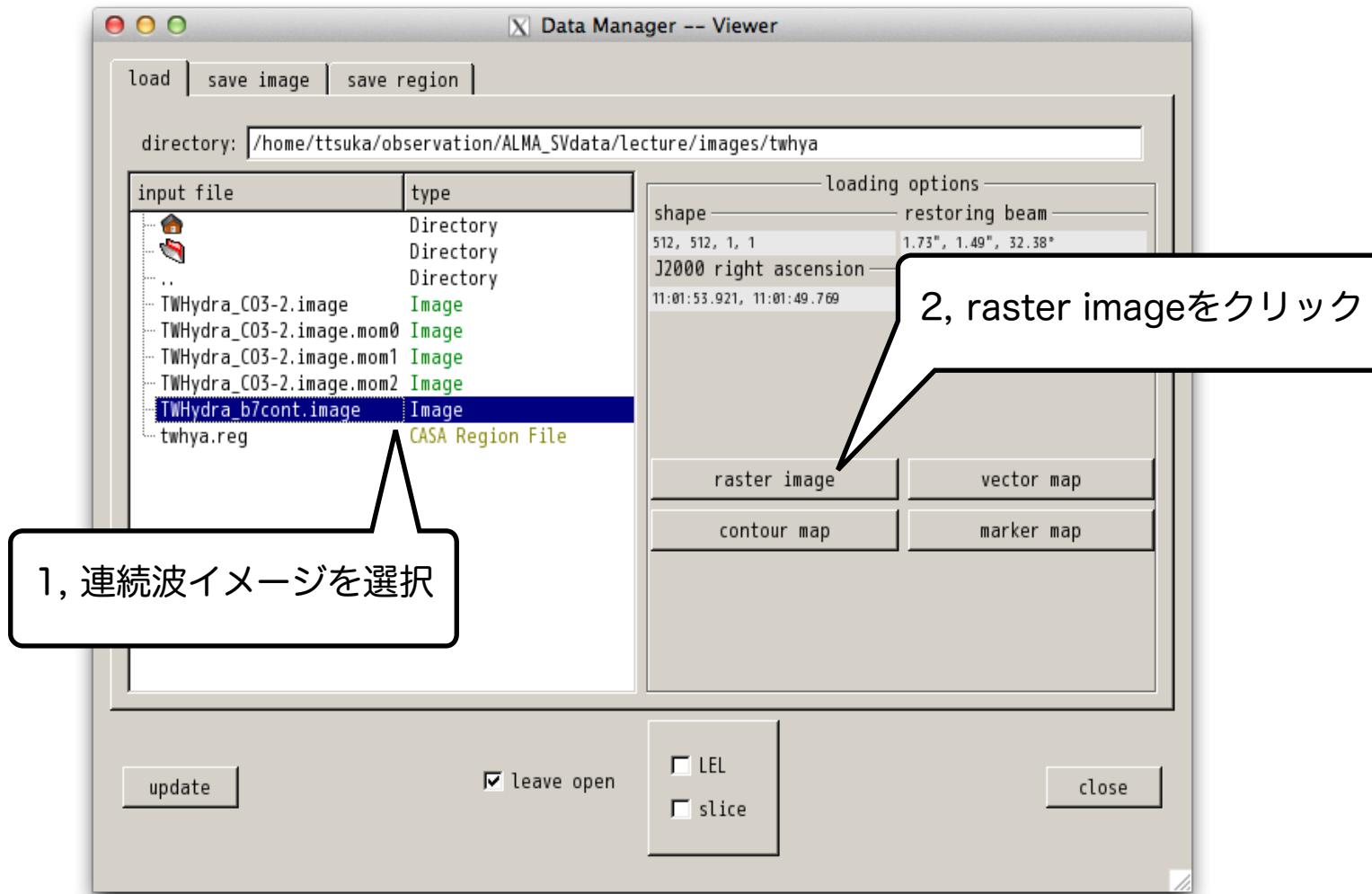
## ポリゴン領域指定

スペクトルやヒストグラムを書く領域の指定。連続クリックで領域を指定する。ダブルクリックで指定を終了。ESCキーで消える。

**連續波マップを見る**

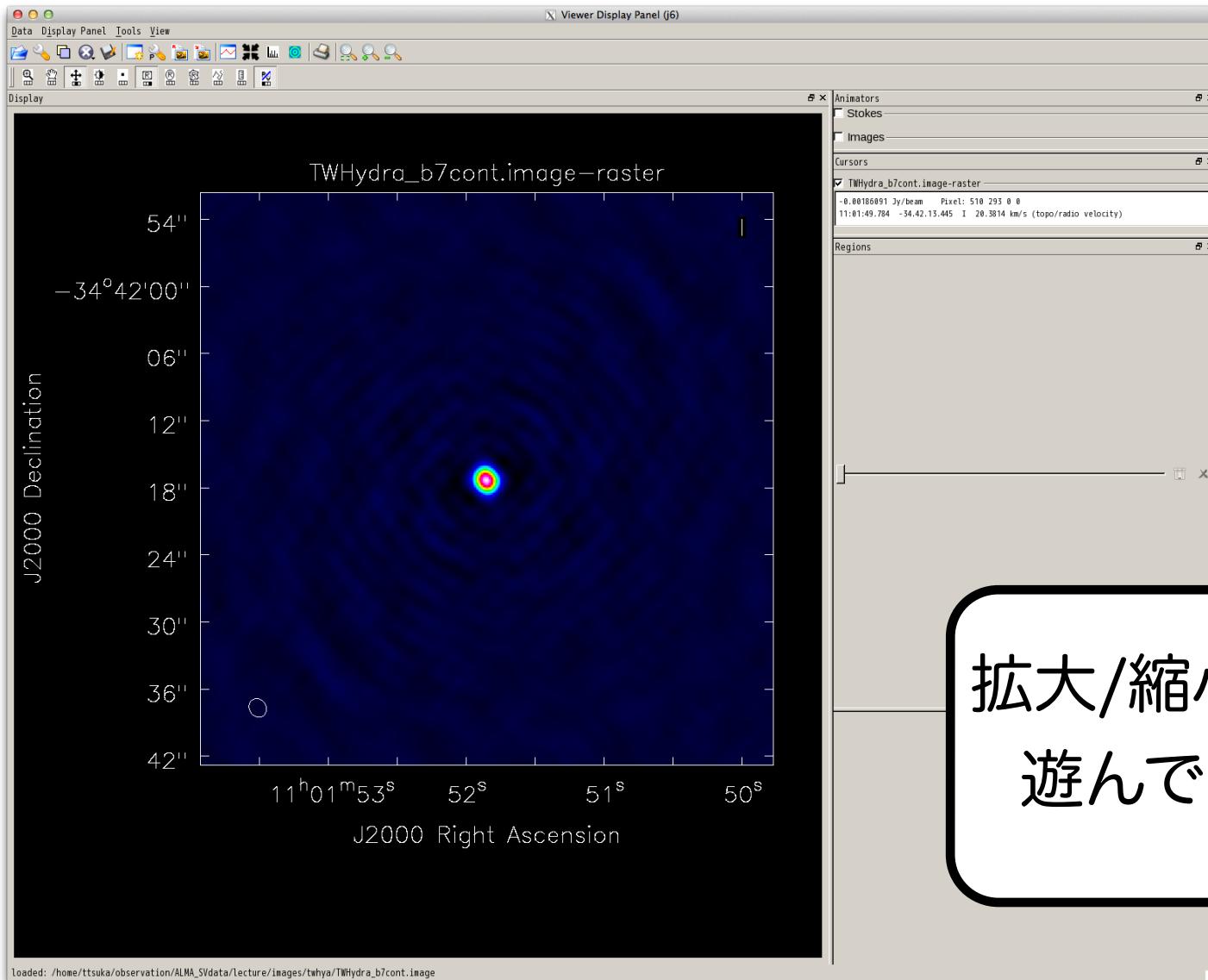
# イメージファイルの読み込み

- 1, CASA viewerを起動
  - # casaviewer <return>



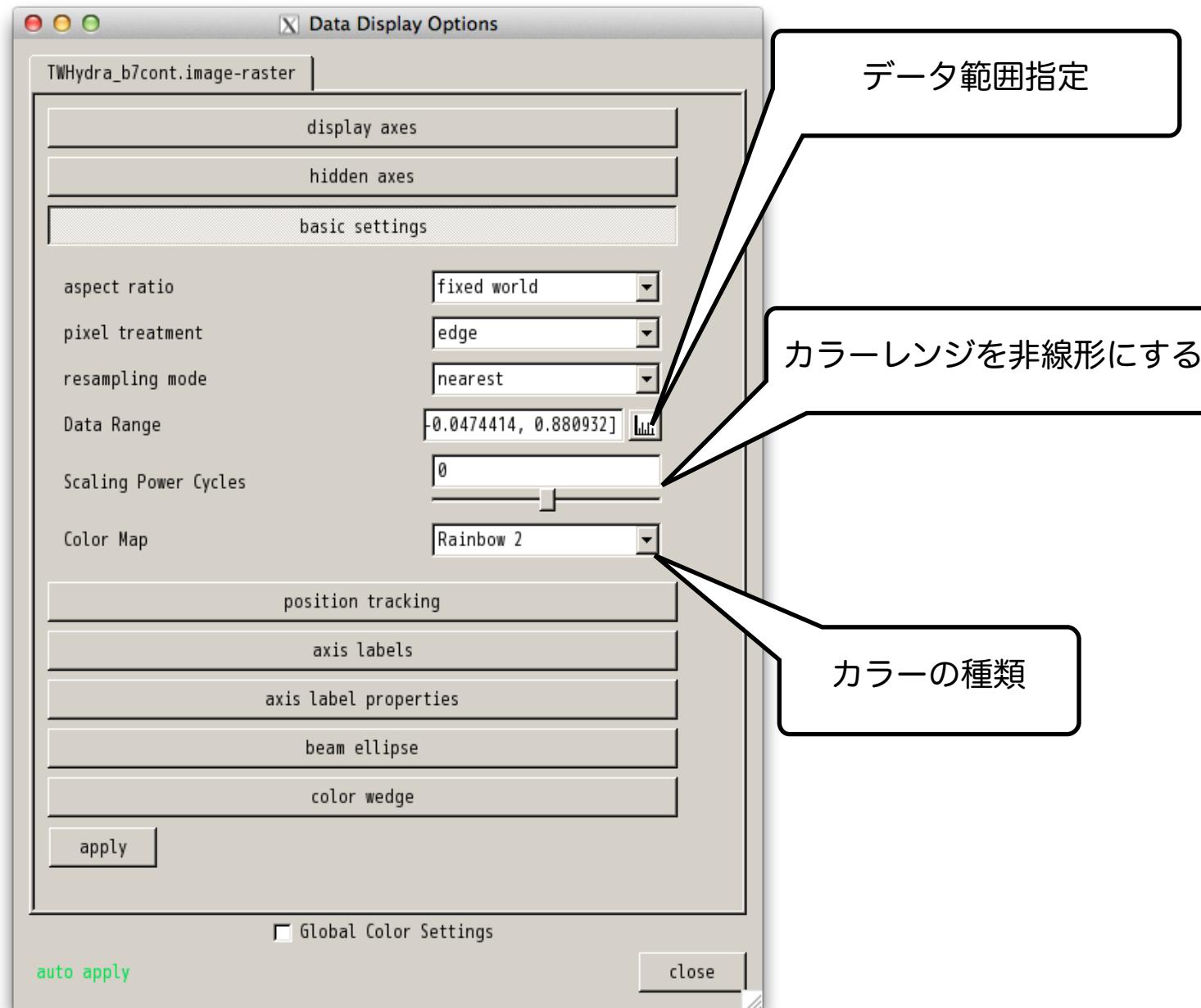
# イメージファイルの読み込み

- 1, CASA viewerを起動
  - # casaviewer <return>

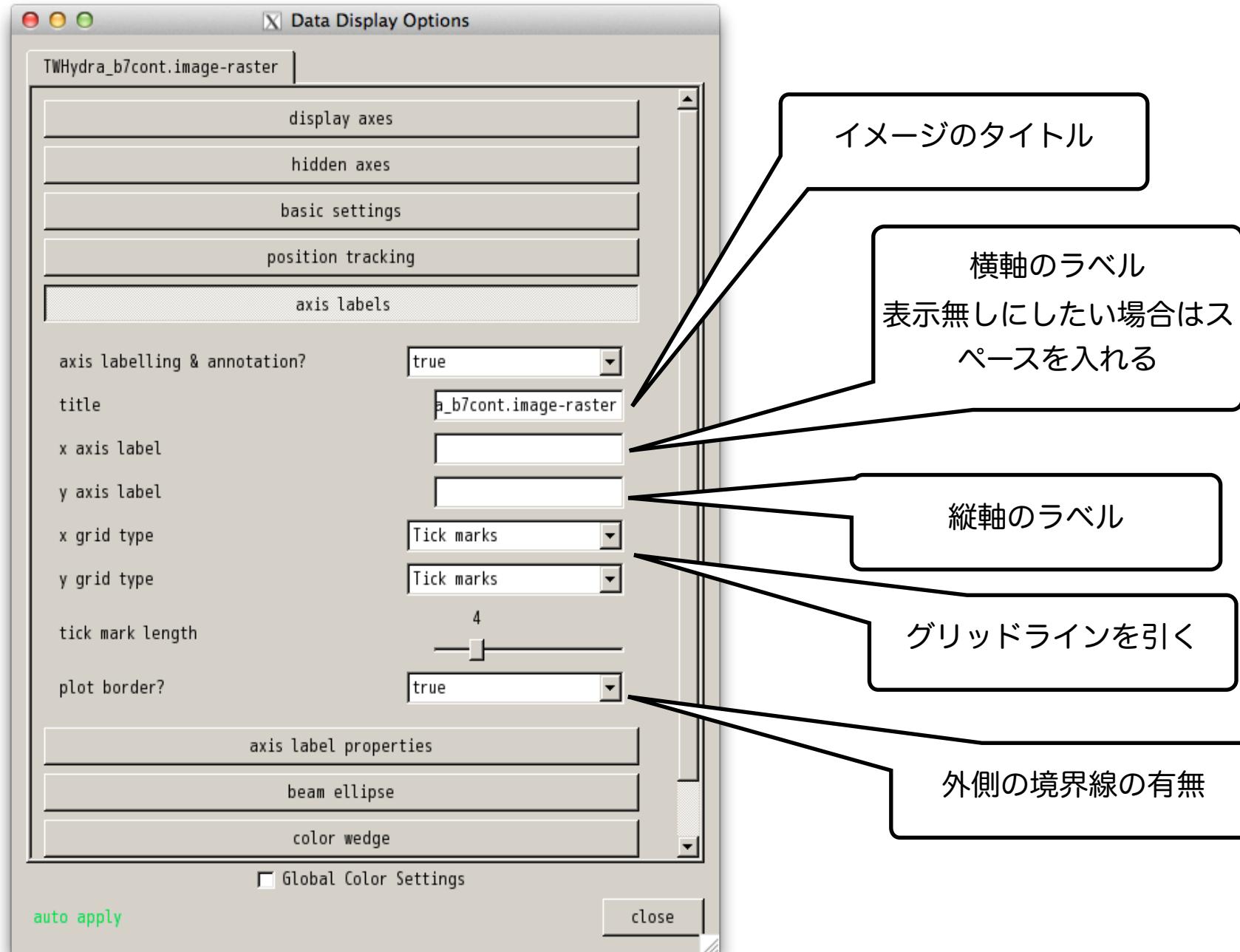


# イメージの調整

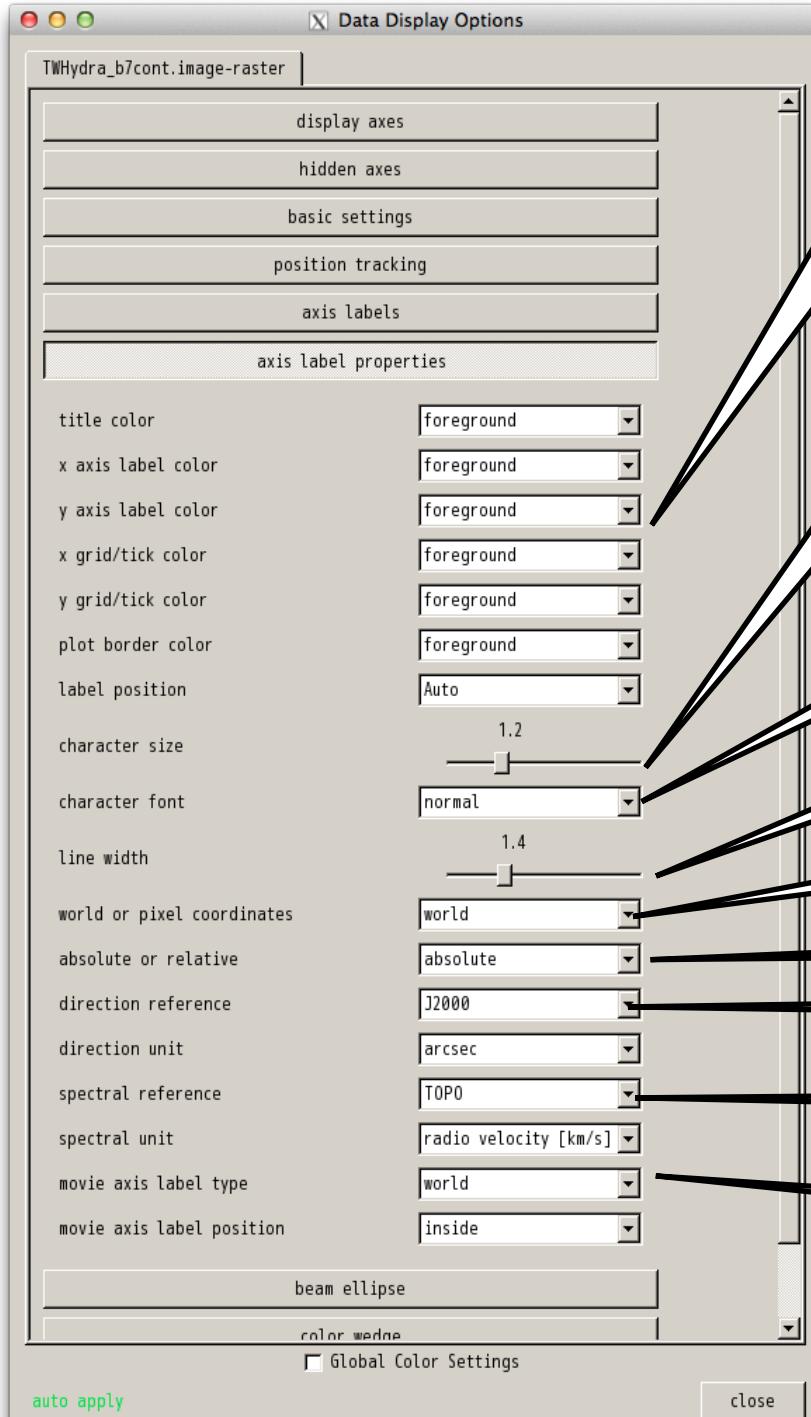
- ツールバーのスパンナ(?)アイコン



# イメージの調整



# イメージの調整



グリッドやラベルのカラーの種類

ラベルの文字サイズ

ラベルのフォント

フォントの太さ

座標かピクセルか

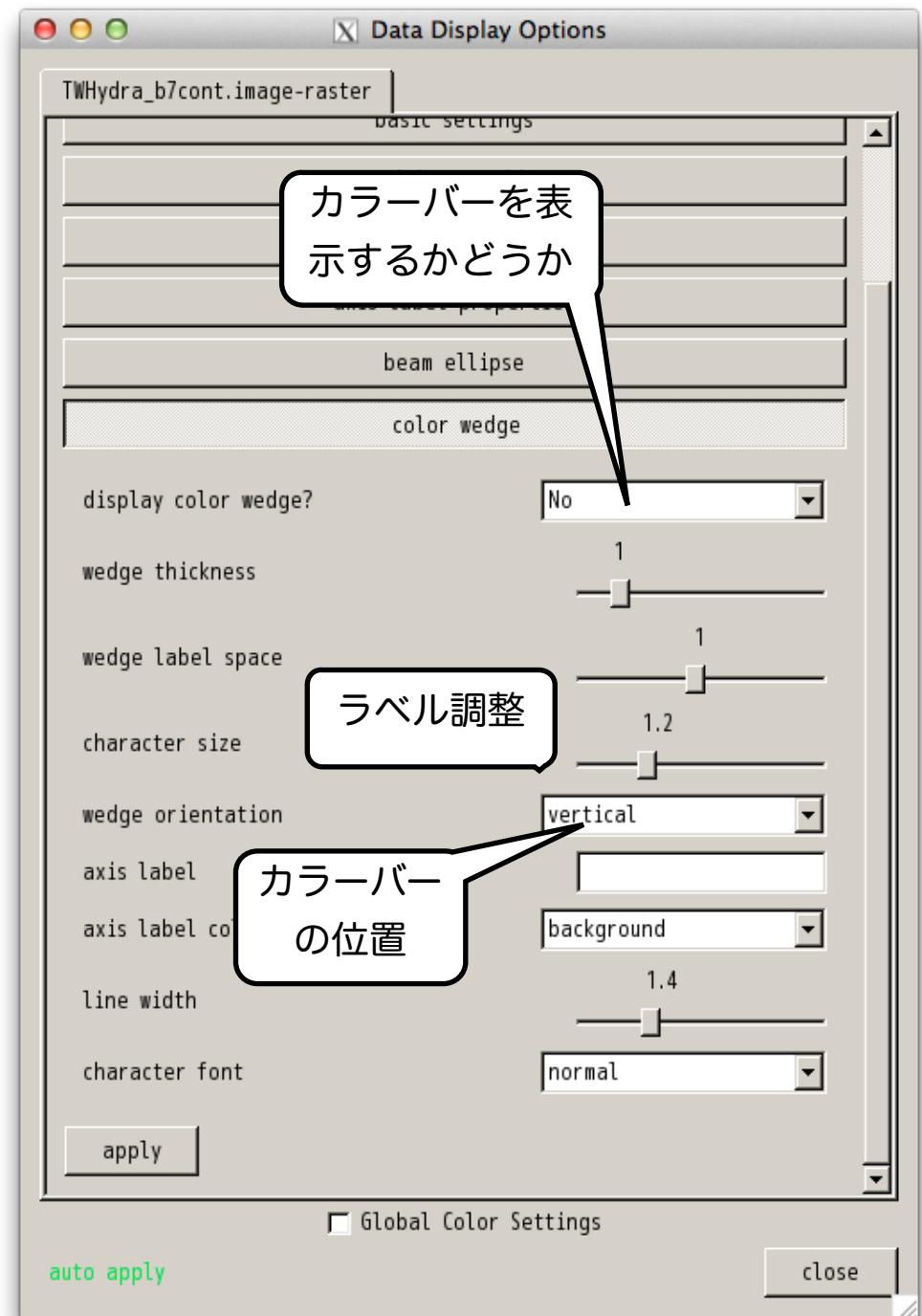
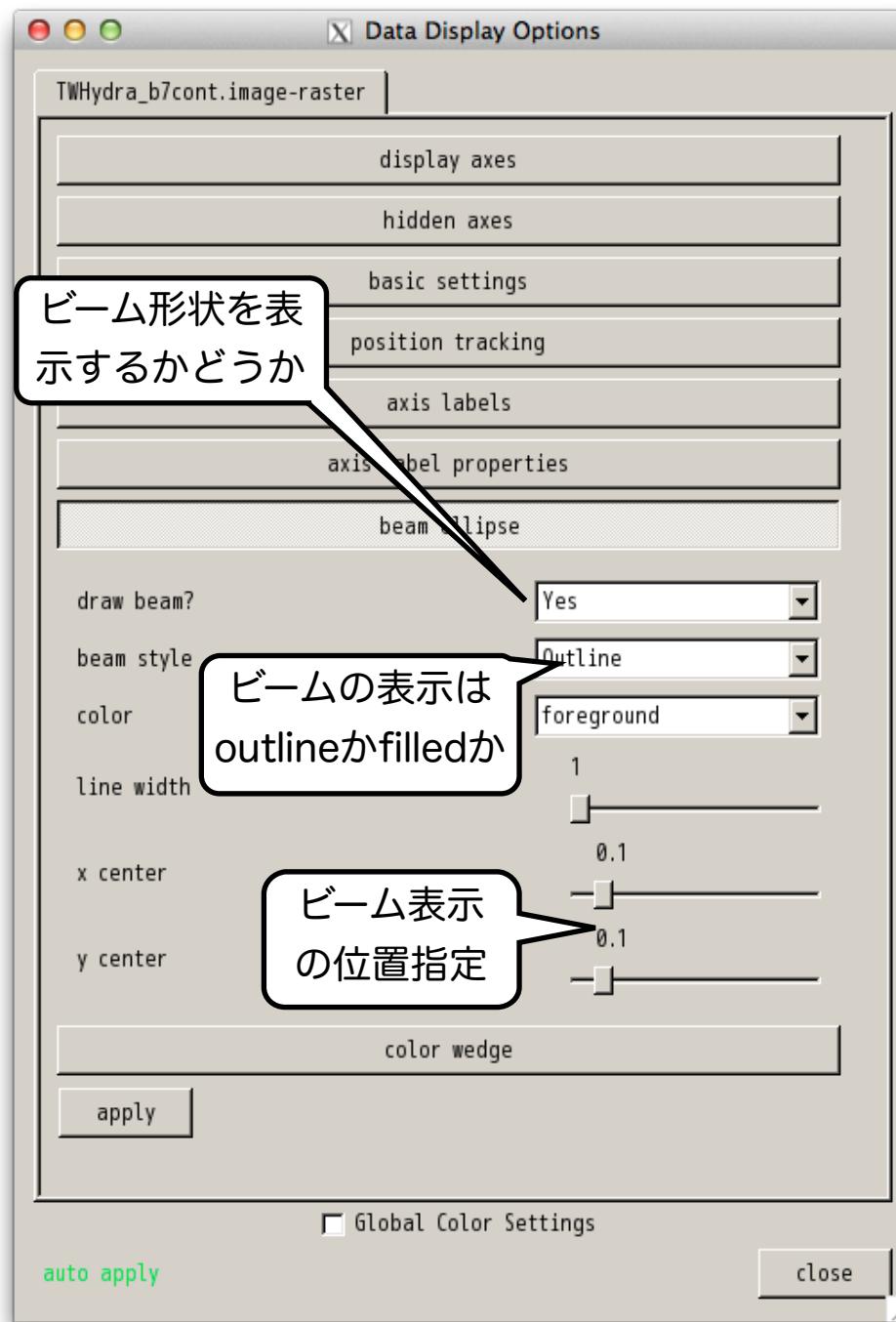
座標の絶対値/相対値

天球座標系指定

速度フレームの指定

速度軸の単位や表示方法

# イメージの調整



# フラックスの測り方

## 1. 扱い易いレベル まで拡大/縮小

2.領域指定ツールを選択(ここでは橢円)

### 3. 測りたい領域を指定する

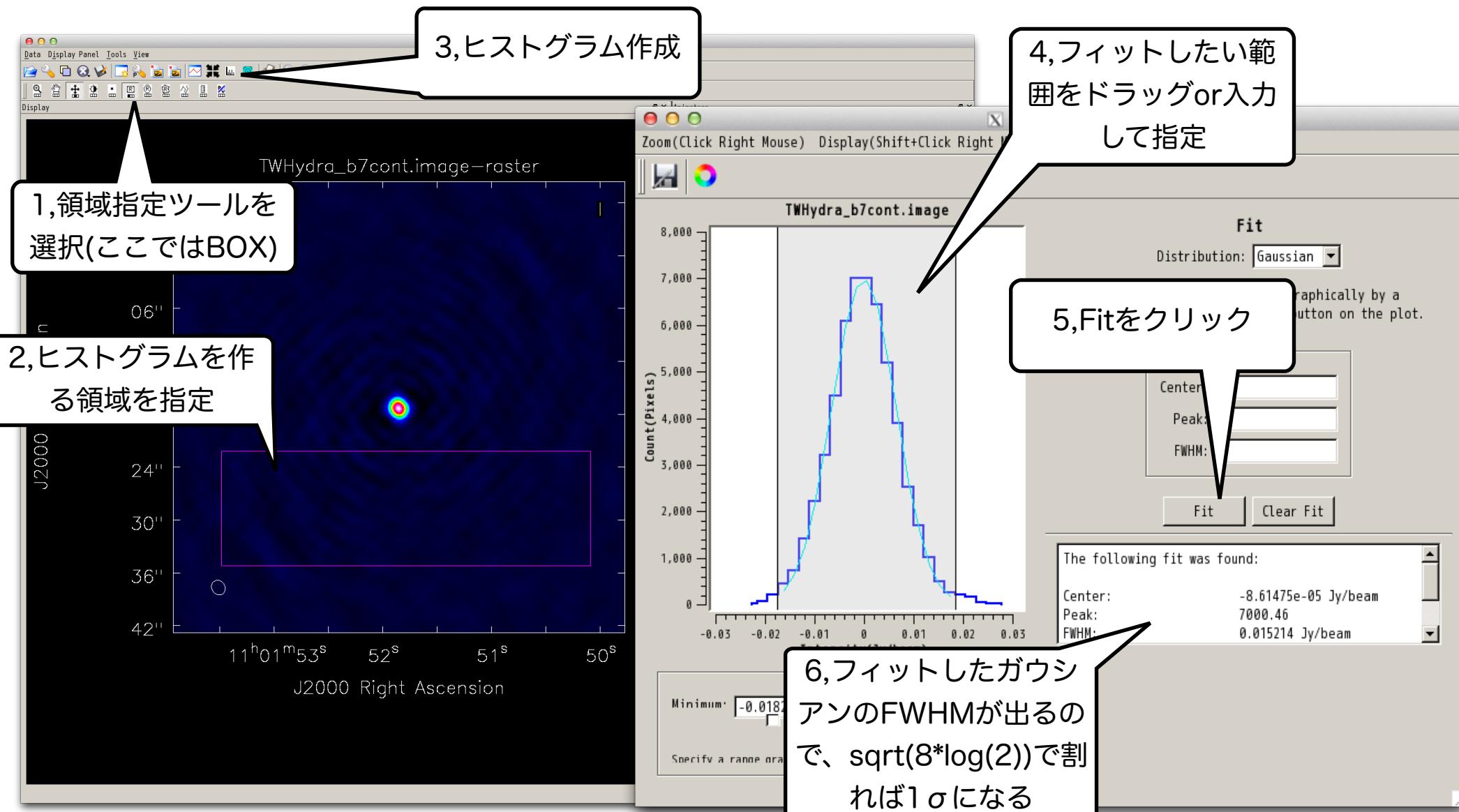
4. 指定した領域ボックス  
内でダブルクリック

5, viewerを起動した  
ターミナルに色々な統  
計値が出力される

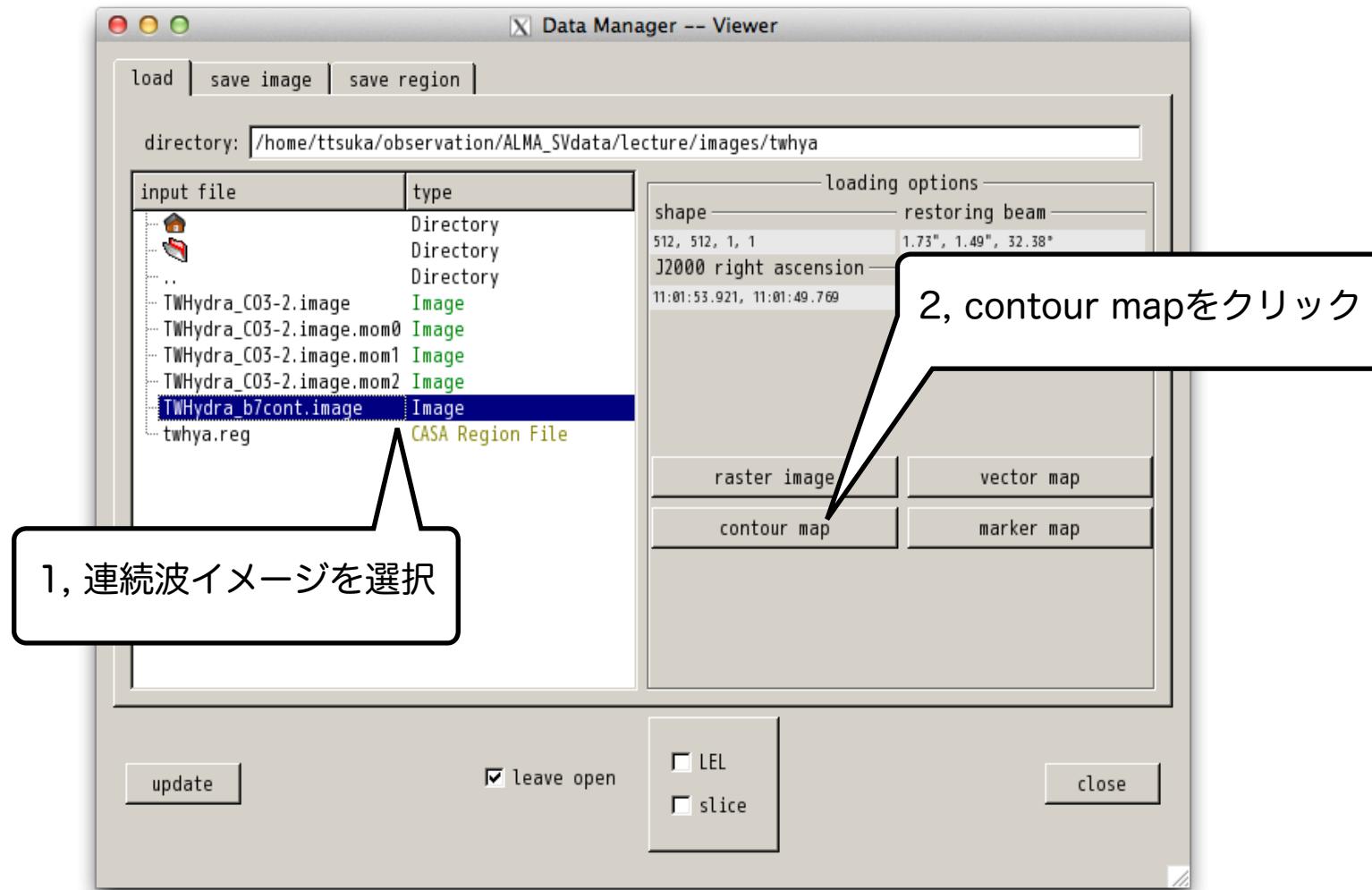
```
[screen 1] [Power] — ssh — 84x22
(TWHydra_C03-2.image)
-----
(TWHydra_C03-2.image)
-----
[1:orion]:tttsuka$          [~/observation/ALMA_SVdata/lecture/images/twhya]
[1:orion]:tttsuka$ casaviewer [~/observation/ALMA_SVdata/lecture/images/twhya]
[1:orion]:tttsuka$ casaviewer [~/observation/ALMA_SVdata/lecture/images/twhya]
[1:orion]:tttsuka$ casaviewer [~/observation/ALMA_SVdata/lecture/images/twhya]
-----
(TWHydra_b7cont.image)
      Stokes      Velocity      Frame      Doppler      Frequency
      I    20.3814km/s    LSRK      RADIO  3.50845e+11
BrightnessUnit      BeamArea      Npts      Sum      FluxDensity
      Jy/beam    292.215     1499  2.989718e+02  1.023123e+00
      Mean        Rms      Std dev      Minimum      Maximum
      1.994475e-01  3.036415e-01  2.290280e-01 -4.584175e-02  8.809323e-01
      region count
      1
-----
```

# ノイズレベルの測り方

1. emission freeな領域でRMSを測る方法(前ページ参照)
2. ヒストグラムのガウシアンフィットを用いる方法



# コントアを引く



# コントアを引く

3,イメージ調整パネルを開く

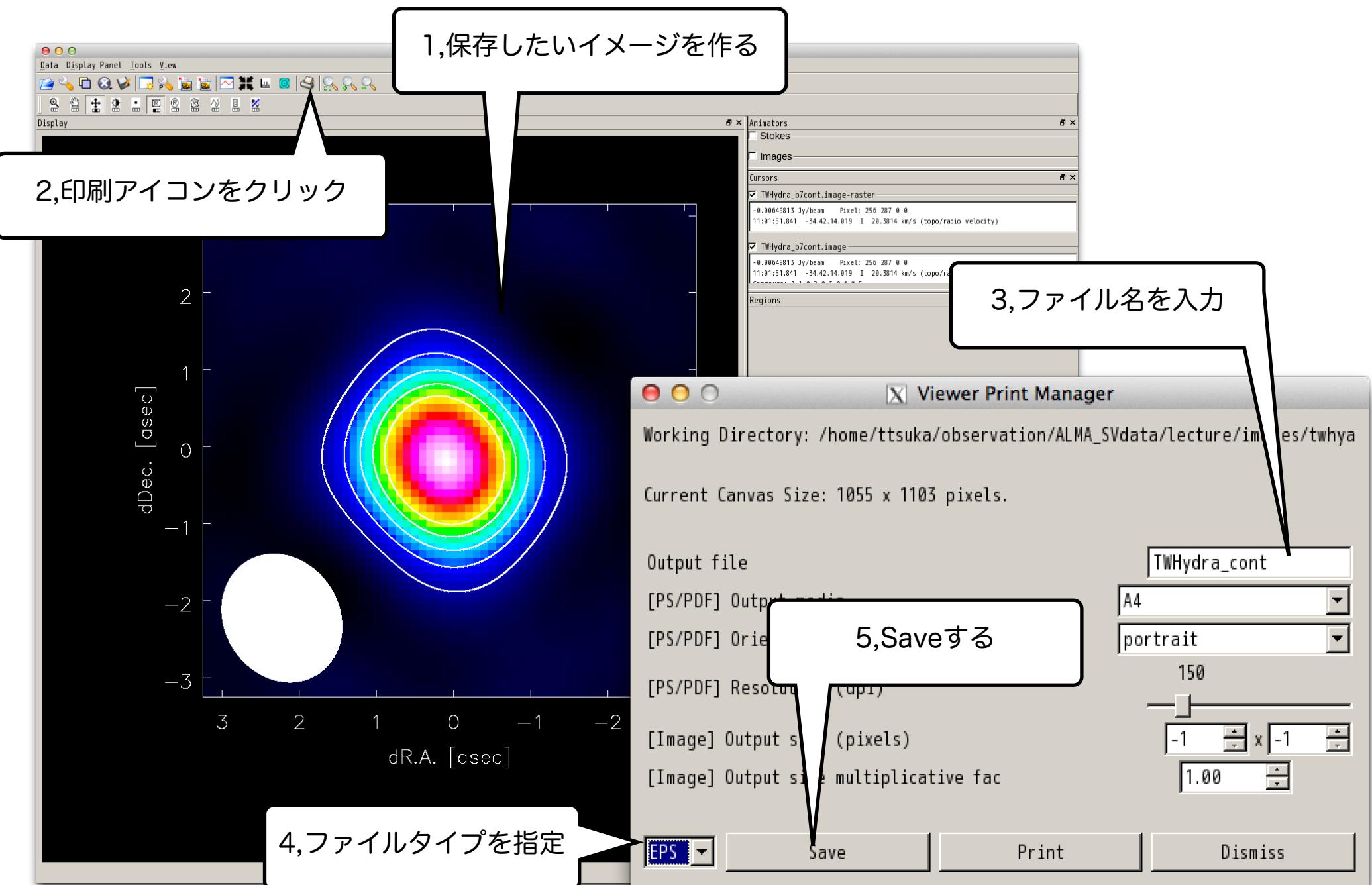
4,コントアにしたイメージタブを選択

5,コントア間隔や線幅、線色などを調整。ここで  
は10,20,30,40,50σで  
引いている

The screenshot shows the 'Data Display Options' dialog box and a 'Viewer Display' window. The dialog box is titled 'Data Display Options' and contains tabs for 'display axes' and 'hidden axes'. It also has sections for 'basic settings' and 'position tracking'. The 'basic settings' section includes options for aspect ratio ('fixed world'), pixel treatment ('edge'), resampling mode ('bilinear'), Relative Contour Levels (set to [10, 20, 30, 40, 50]), Base Contour Level (set to 0), Unit Contour Level (set to 0.01), Line width (set to 2), Dash negative contours? (set to true), Dash positive contours? (set to false), and Line color (set to foreground). The 'position tracking' section has a 'Global Color Settings' checkbox and an 'auto apply' button. The 'close' button is at the bottom right.

The 'Viewer Display' window shows a map centered around J2000 Right Ascension 11<sup>h</sup>01<sup>m</sup>52<sup>s</sup>.3 and Declination -34°42'22". The map title is 'TWHydra\_b7cont.image-raster'. The map displays concentric contour lines representing data values. A small circle is visible near the bottom left. The y-axis ranges from 12'' to 20'' and -34°42'22". The x-axis ranges from 11<sup>h</sup>01<sup>m</sup>52<sup>s</sup>.0 to 52<sup>s</sup>.0. The map is set against a dark blue background.

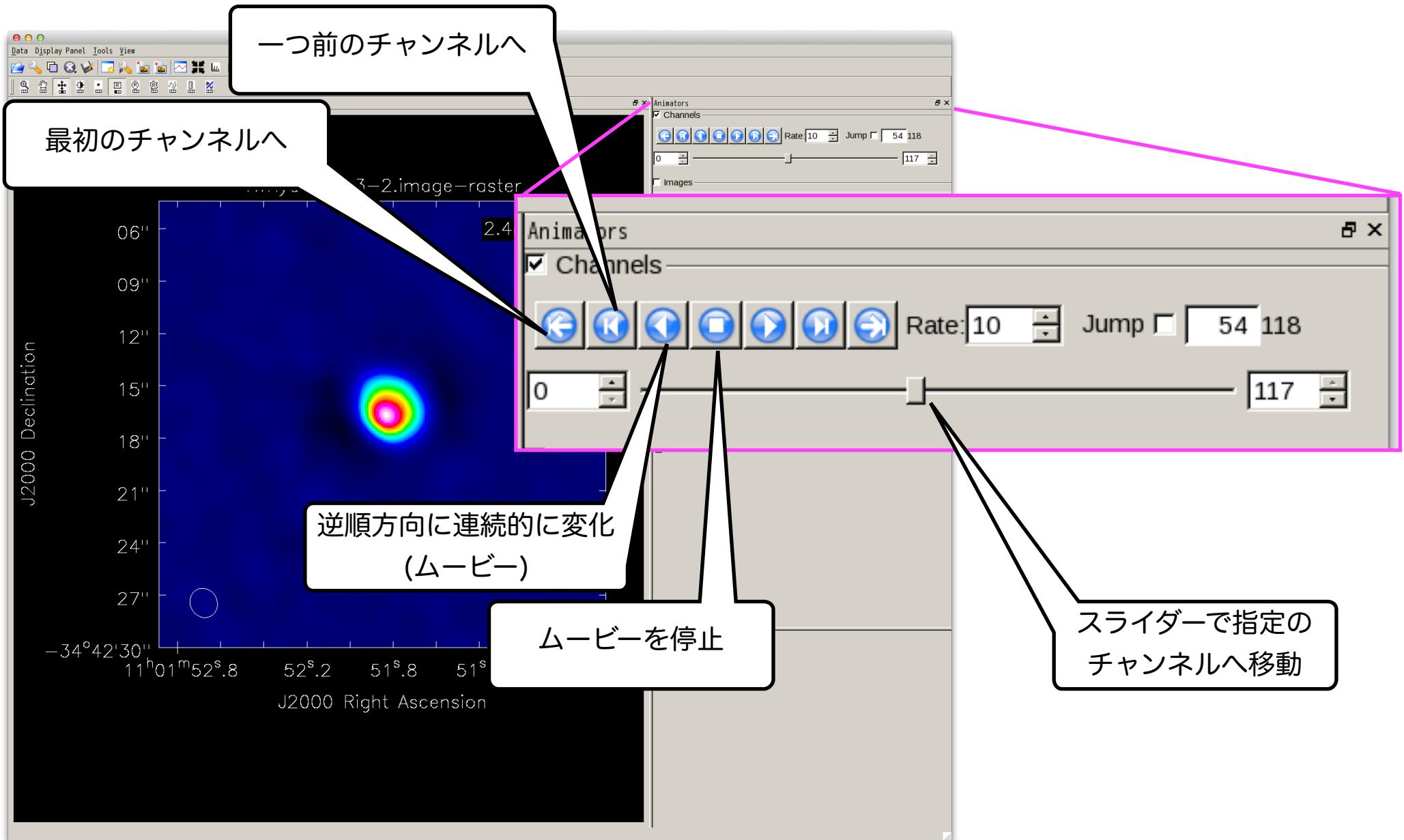
# 画像の保存方法



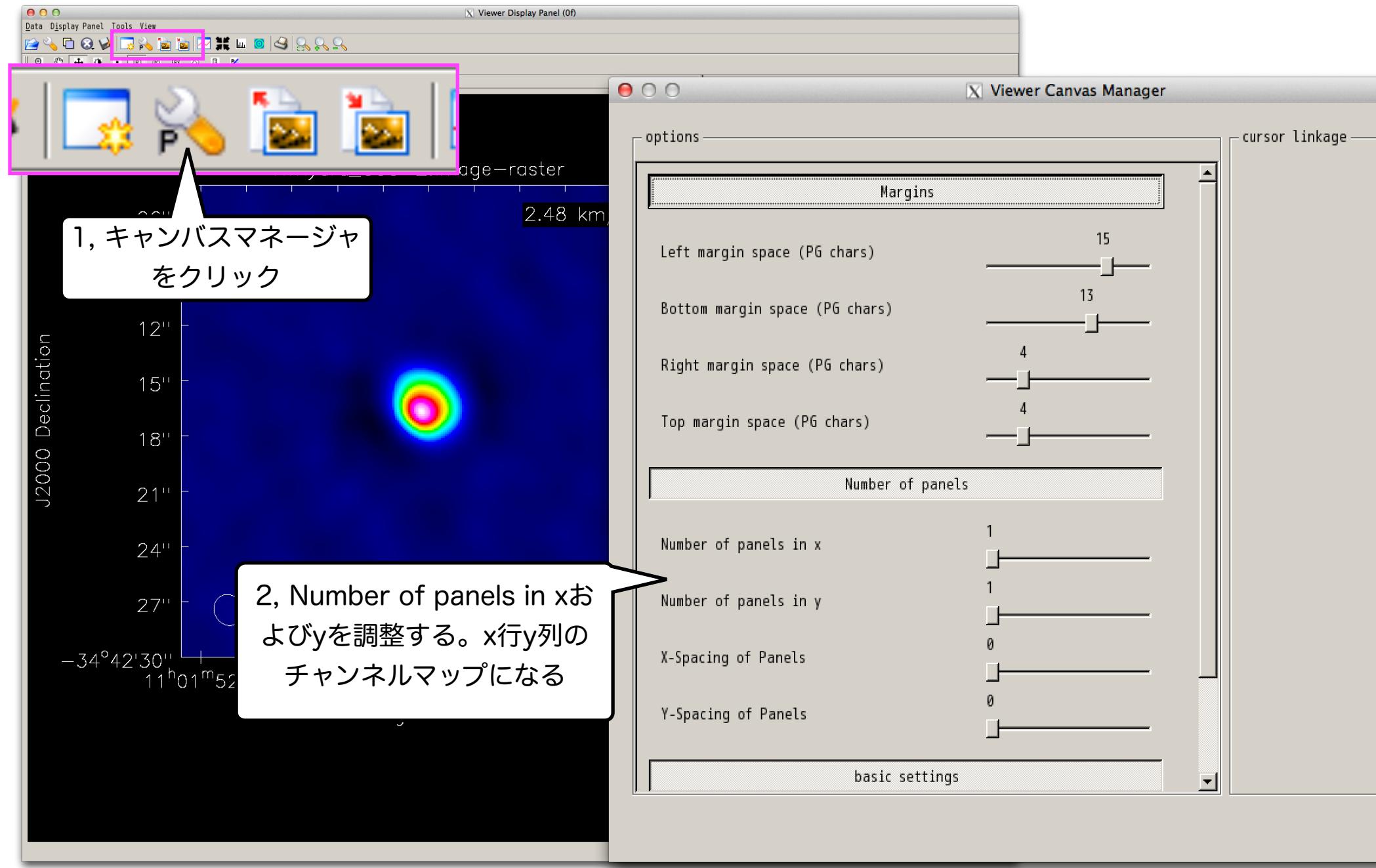
# 3次元マップを見る

# 速度軸のアニメーション

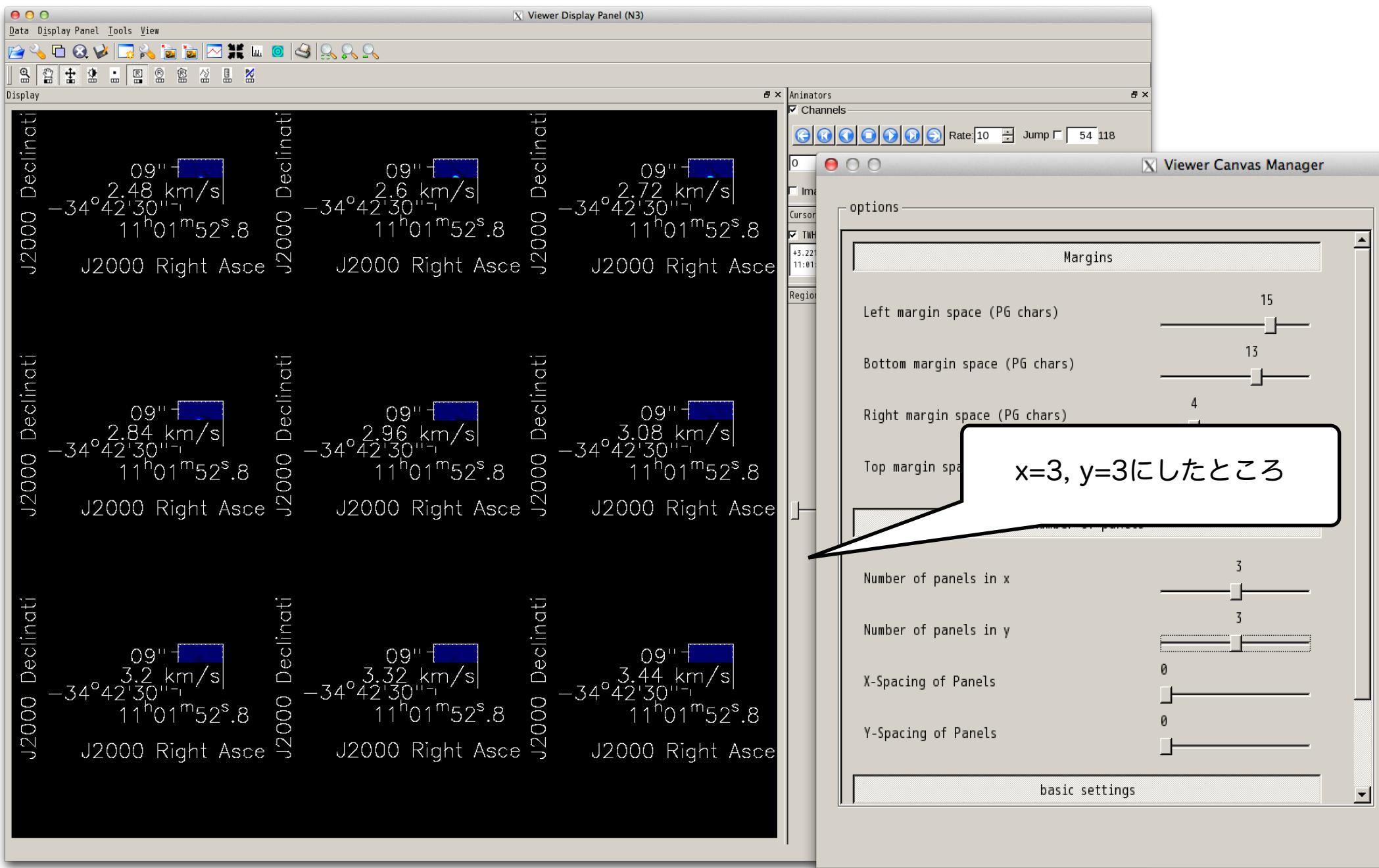
- 3DイメージではChannelsパネルが有効になる



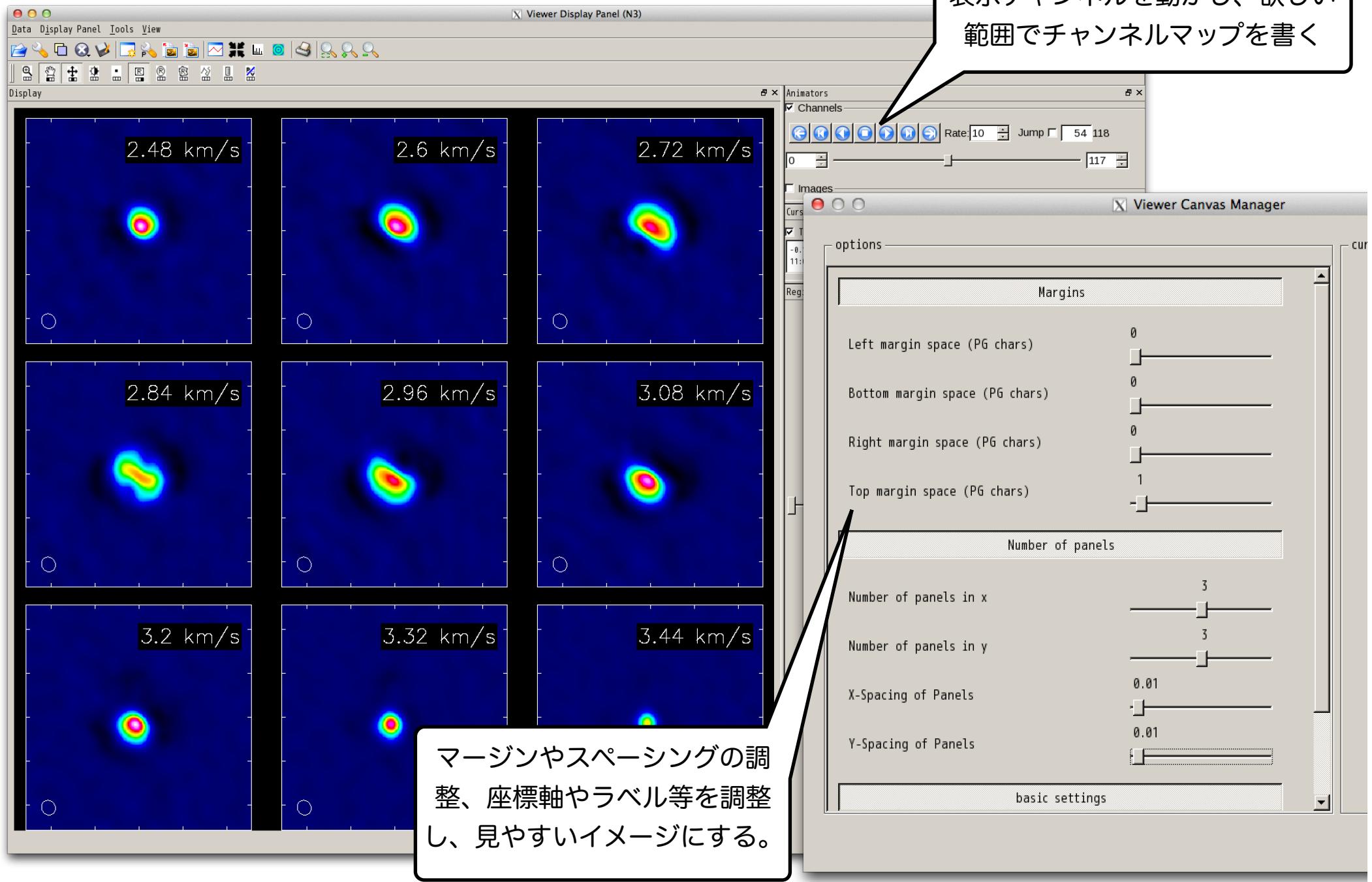
# チャンネルマップの表示方法



# チャンネルマップの表示方法



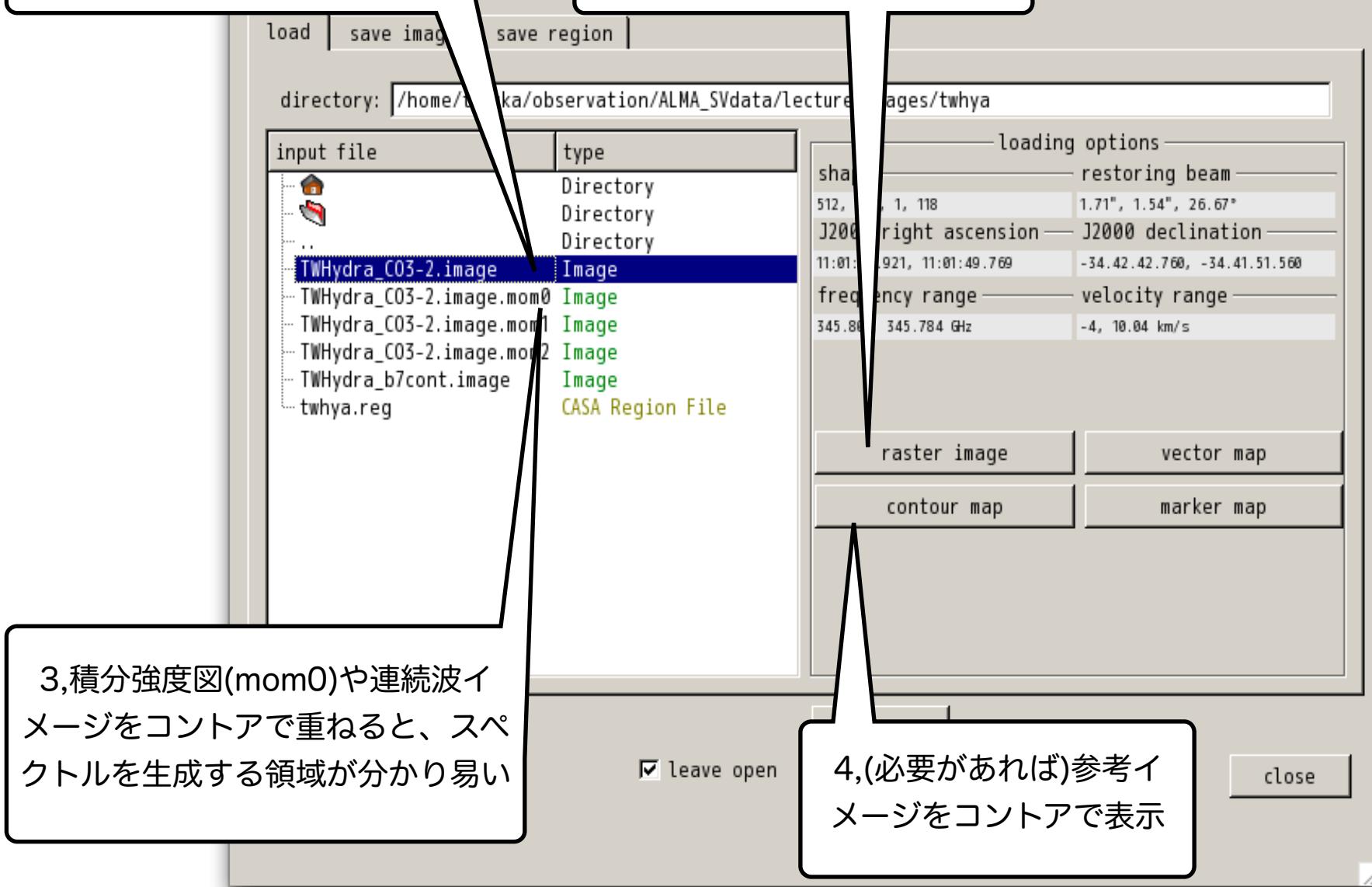
# チャンネルマップの表示方法



# スペクトルプロファイルの作成

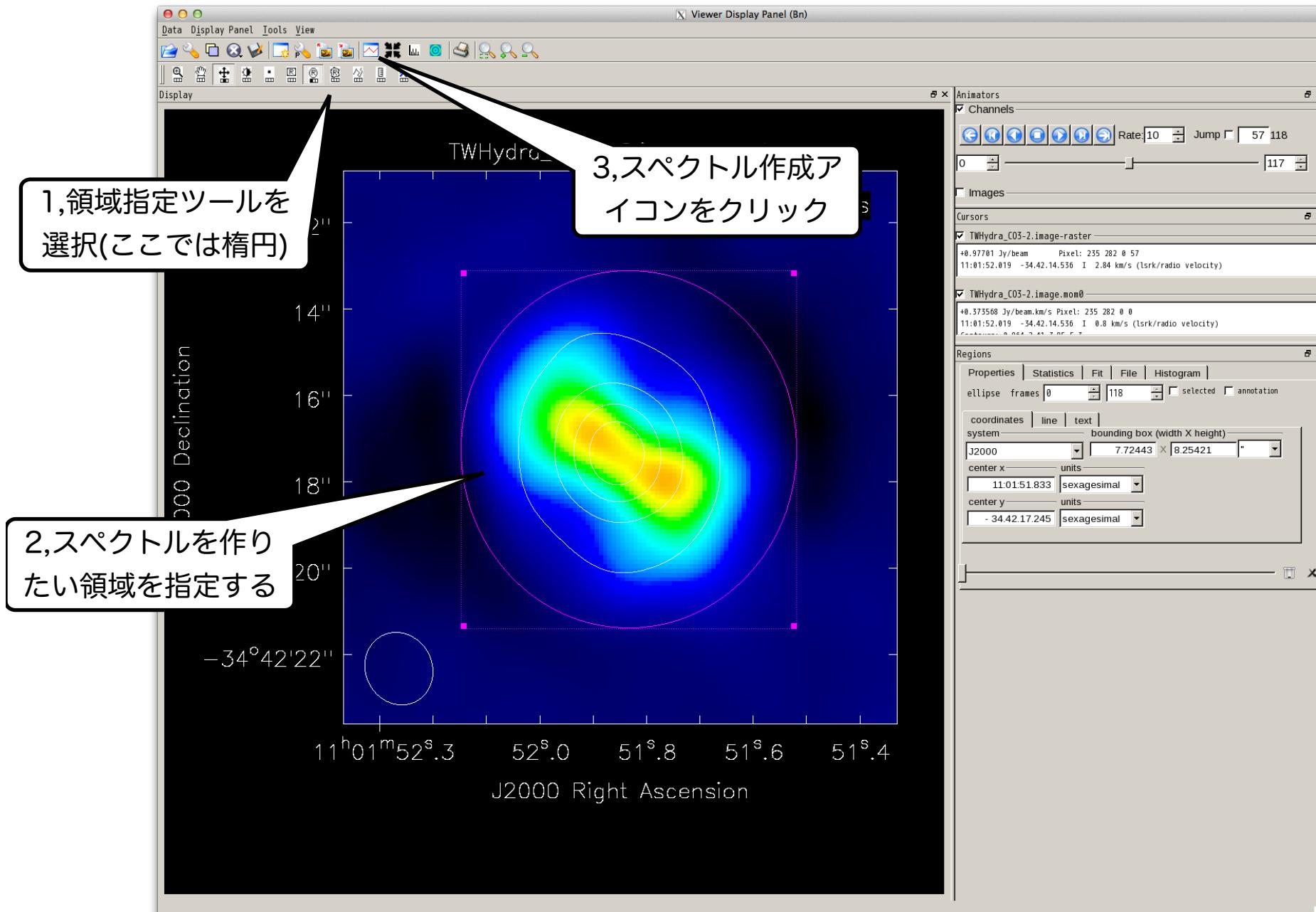
# チャネルマップの読み込み

1,スペクトルのベースになる、  
チャンネルマップを選択

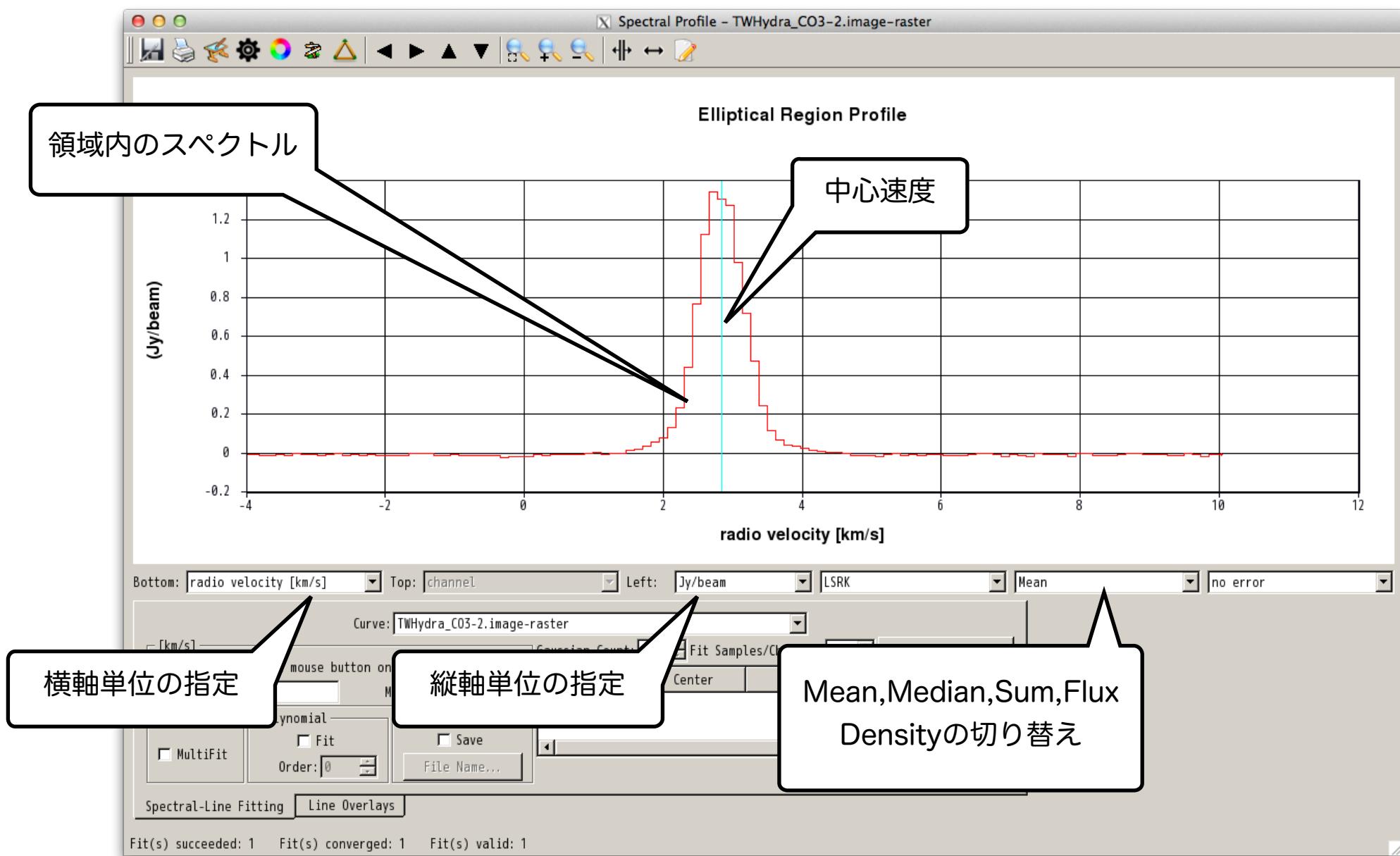


# スペクトル作成の領域を指定

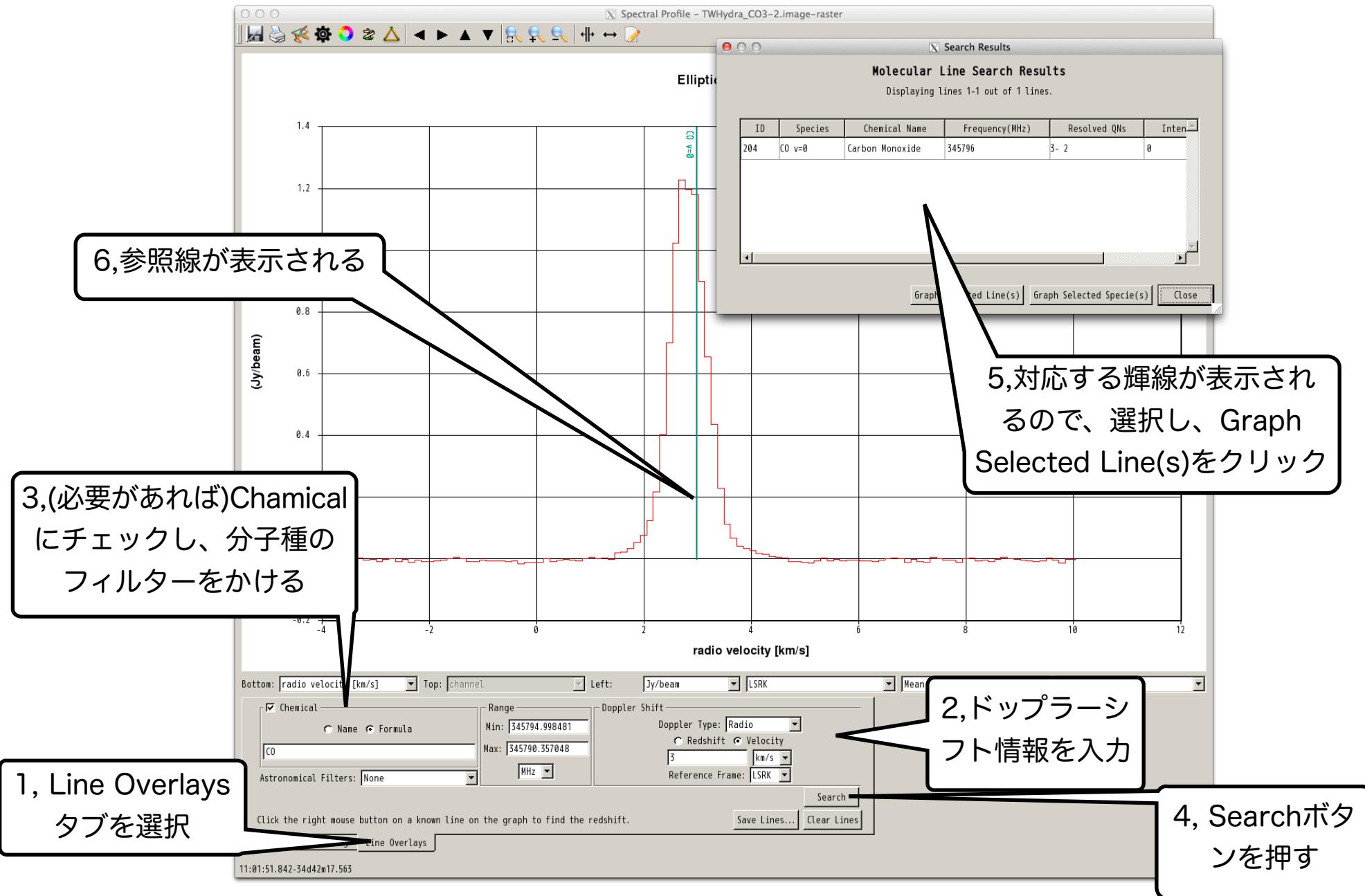
- ・ チャンネルマップ(カラー)+積分強度図(コントア)



# スペクトルプロファイルの画面

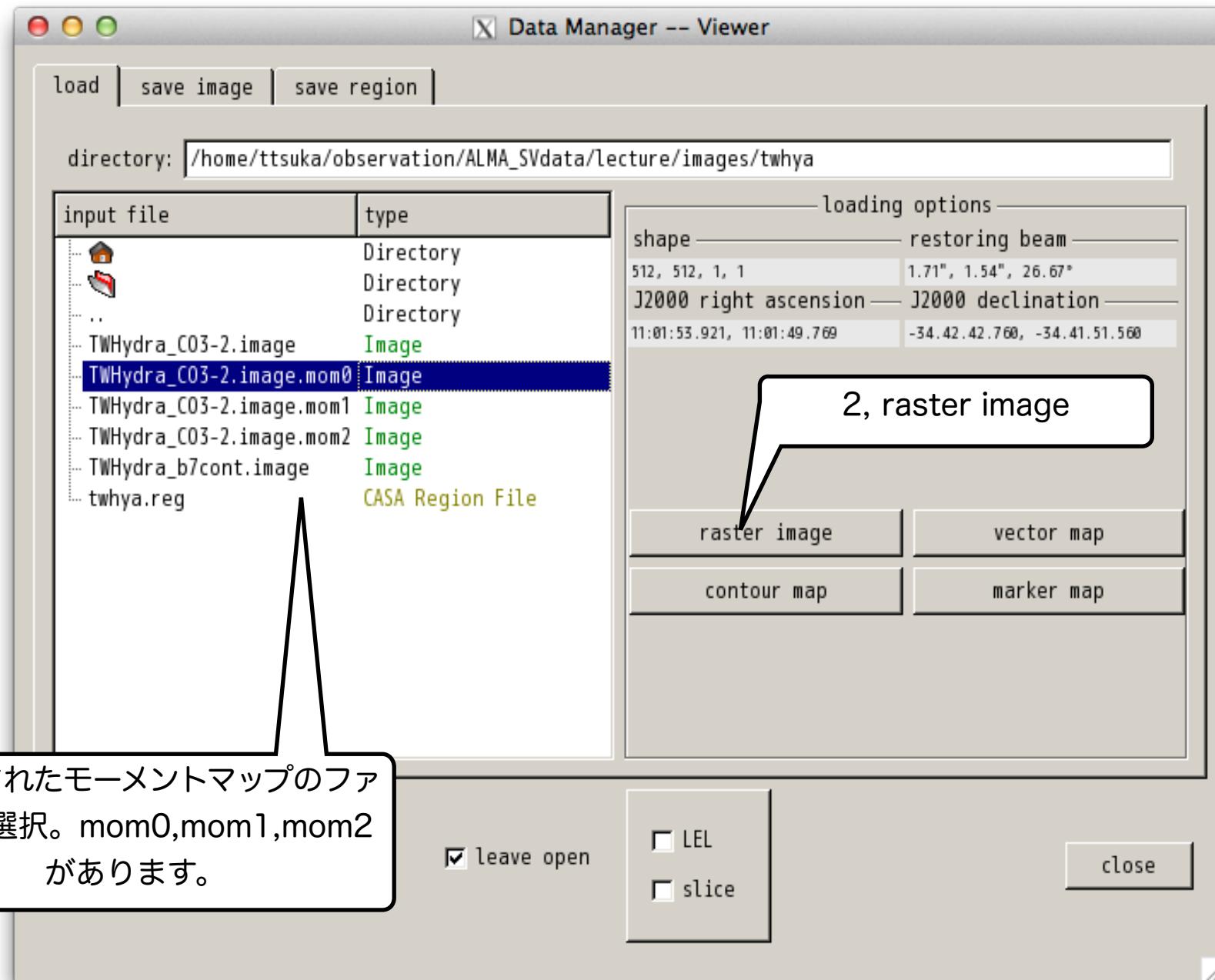


# リファレンススペクトルの表示

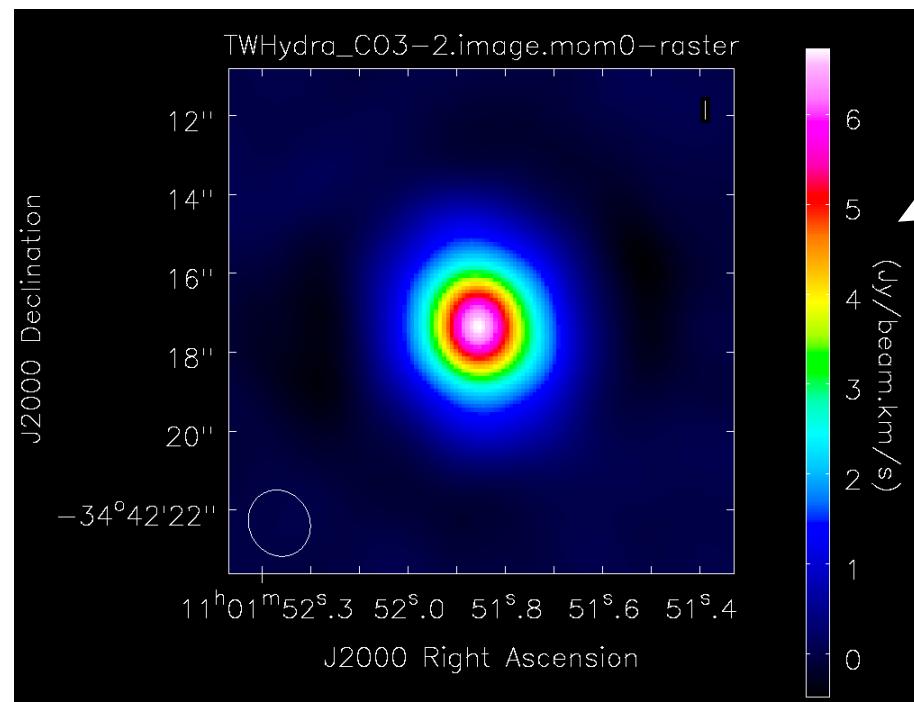


**モーメントマップを見る  
モーメントマップの作成**

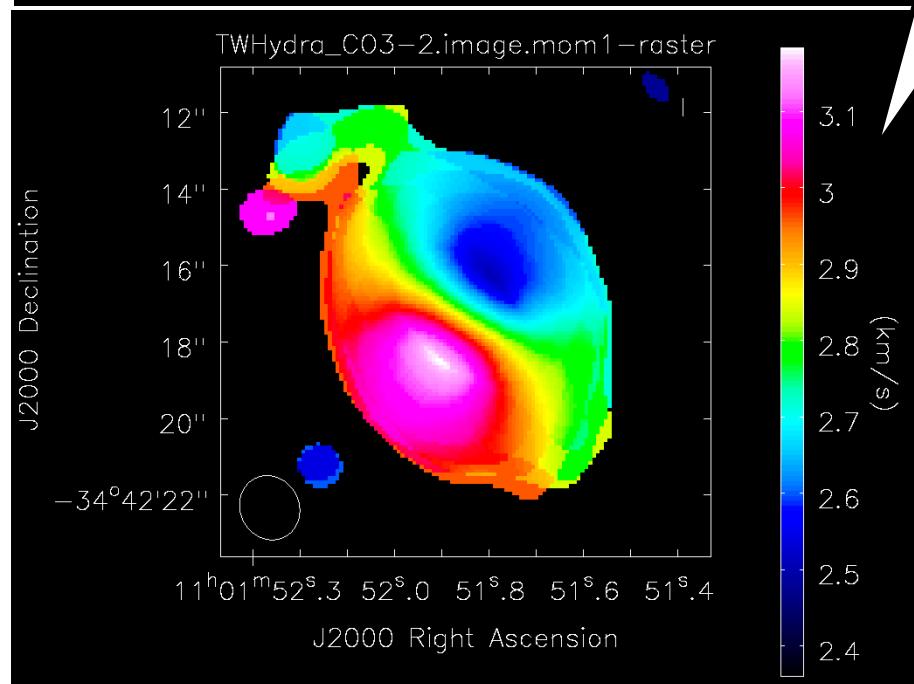
# モーメントマップの読み込み



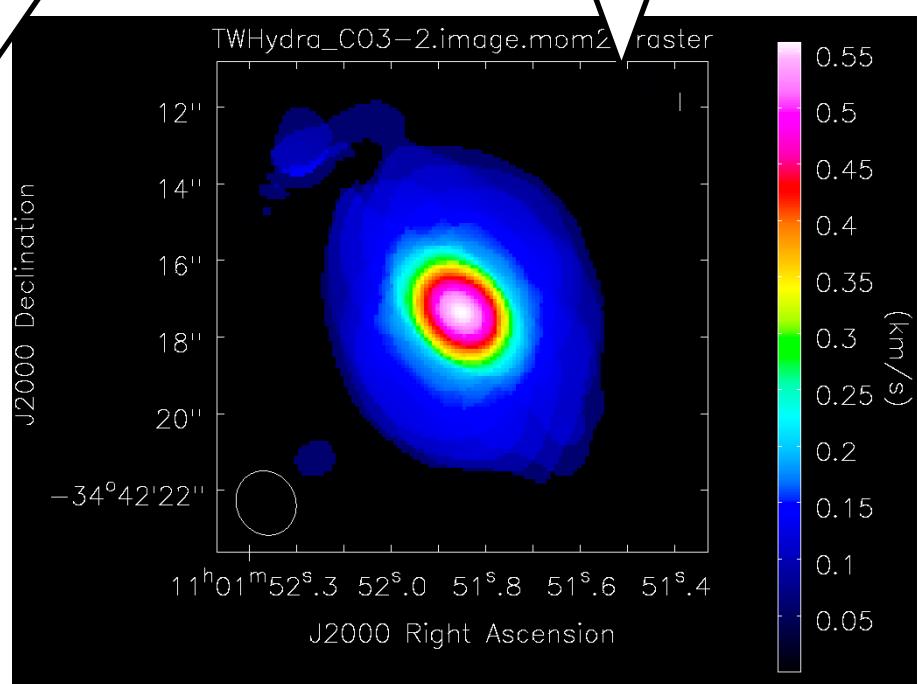
# モーメントマップの表示



moment0(積分強度図)  
単位がJy/beam\*km/s



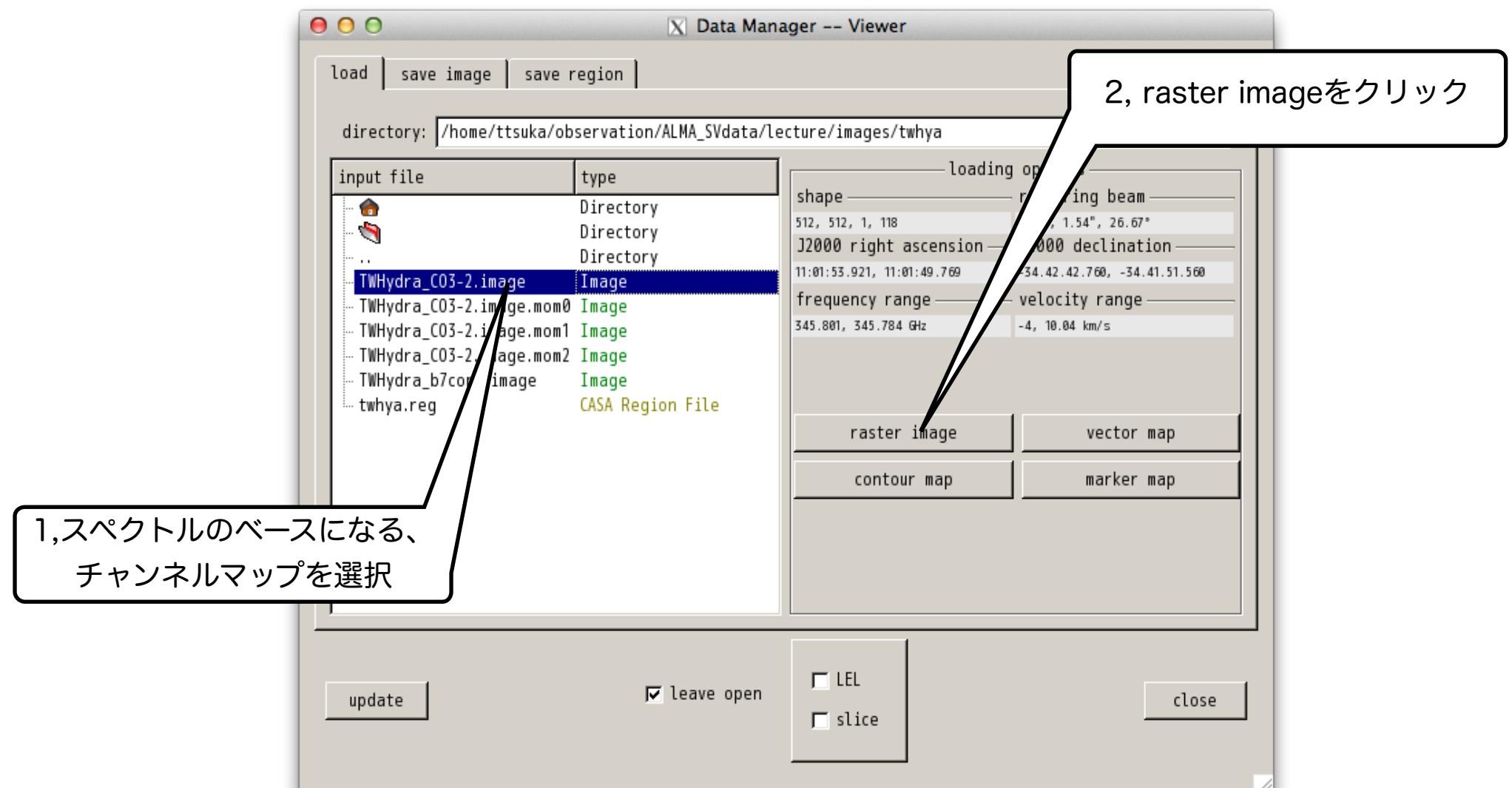
moment1(平均速度場)  
単位がkm/s



moment2(平均分散)  
単位がkm/s

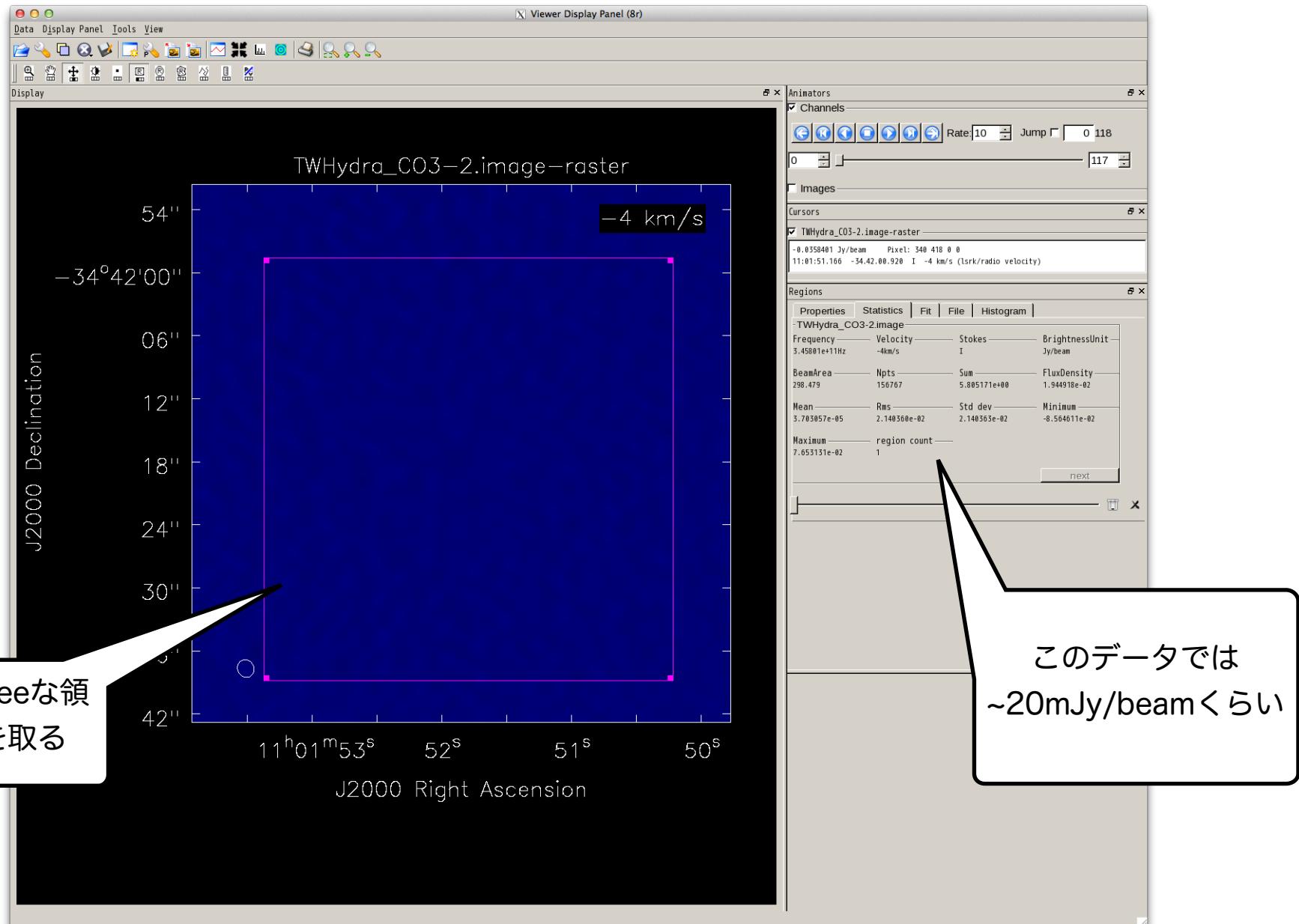
# モーメントマップの作成

- CASAのタスクを使用する方法
  - 同梱したimmoments\_twhya.pyを参照
- CASA viewerを使用した方法



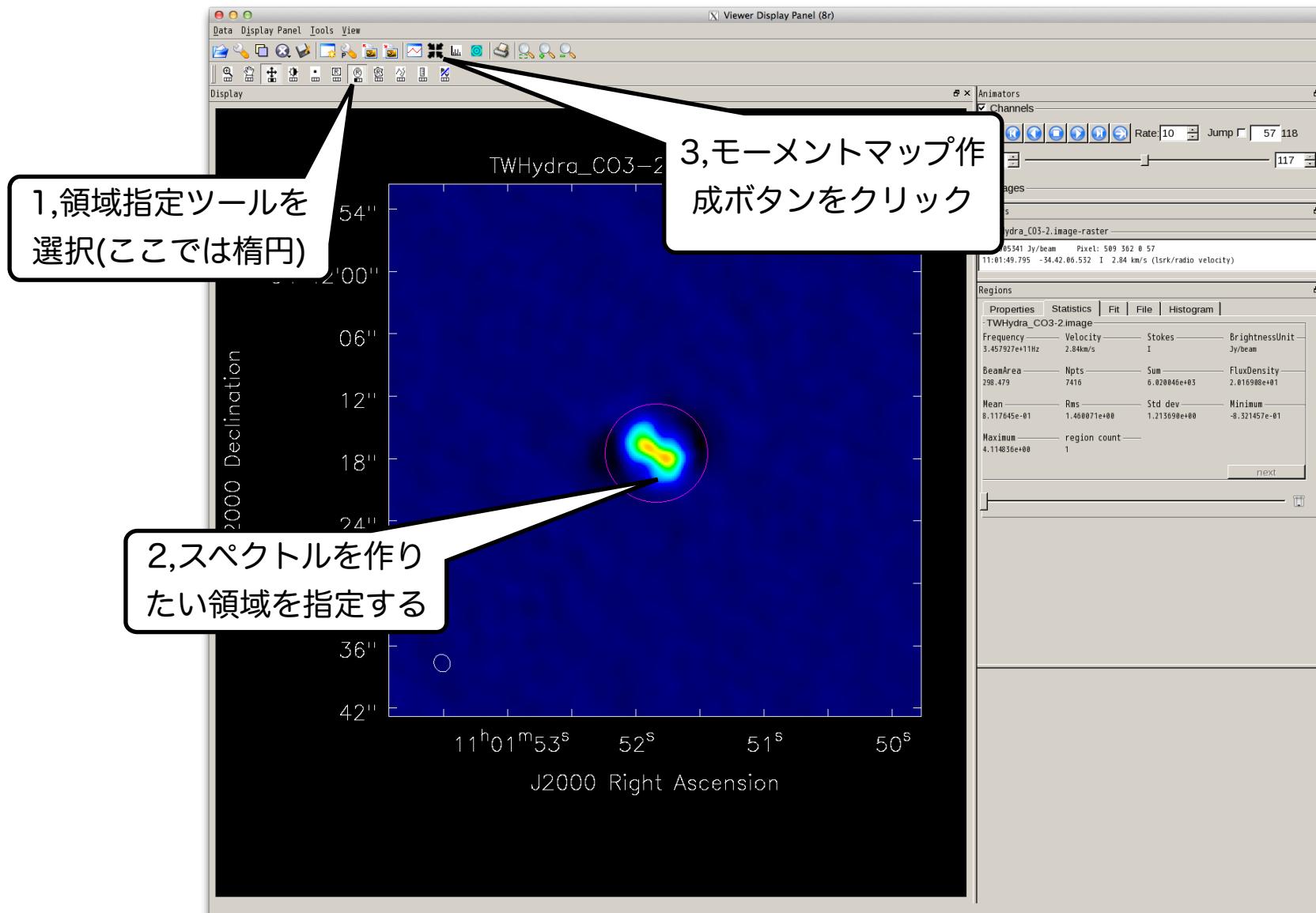
# モーメントマップの作成

- チャンネルマップのノイズレベルを記録しておく

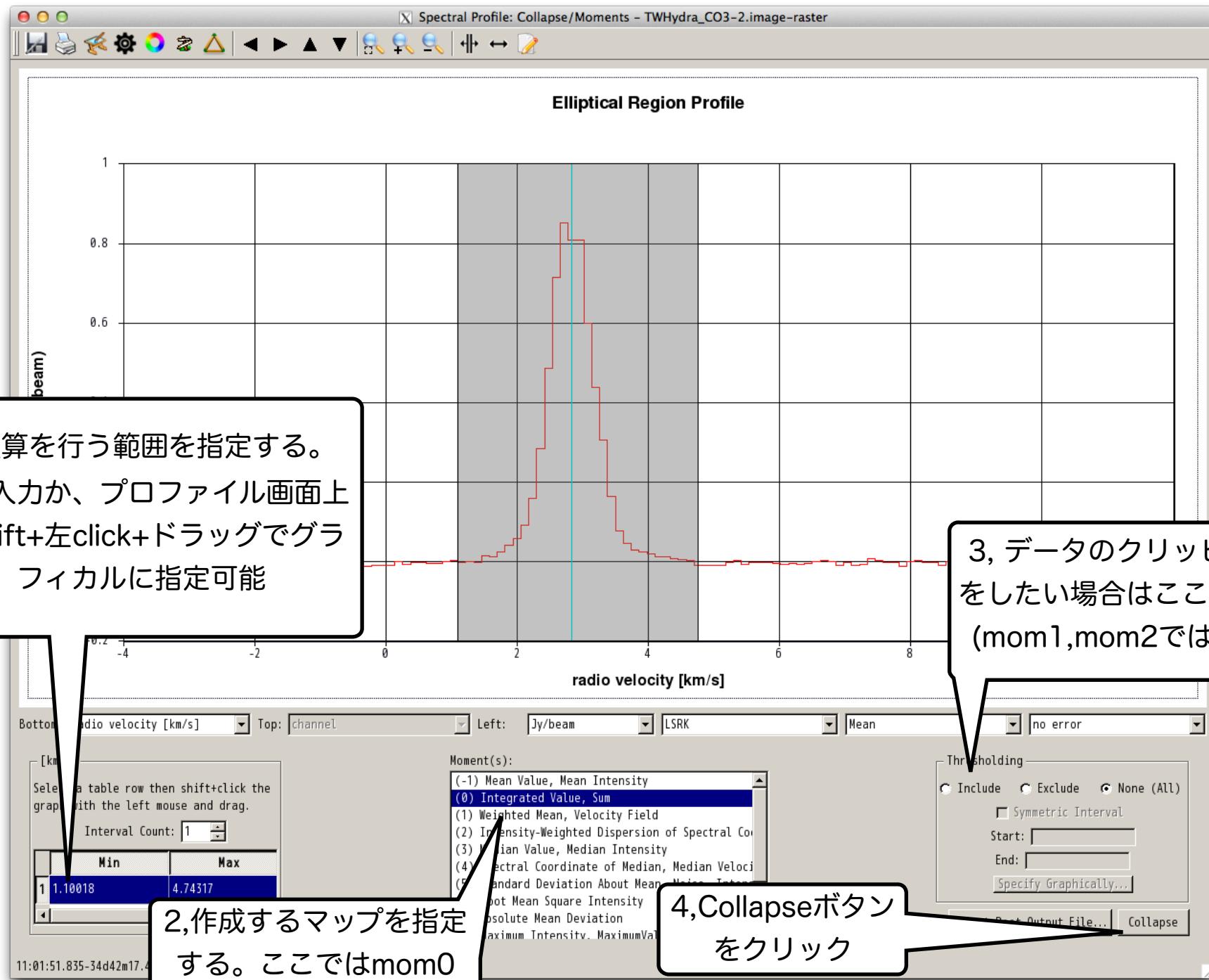


# モーメントマップの作成

- 積分範囲を決める為のスペクトルを表示するので、スペクトルを書くための範囲を指定する

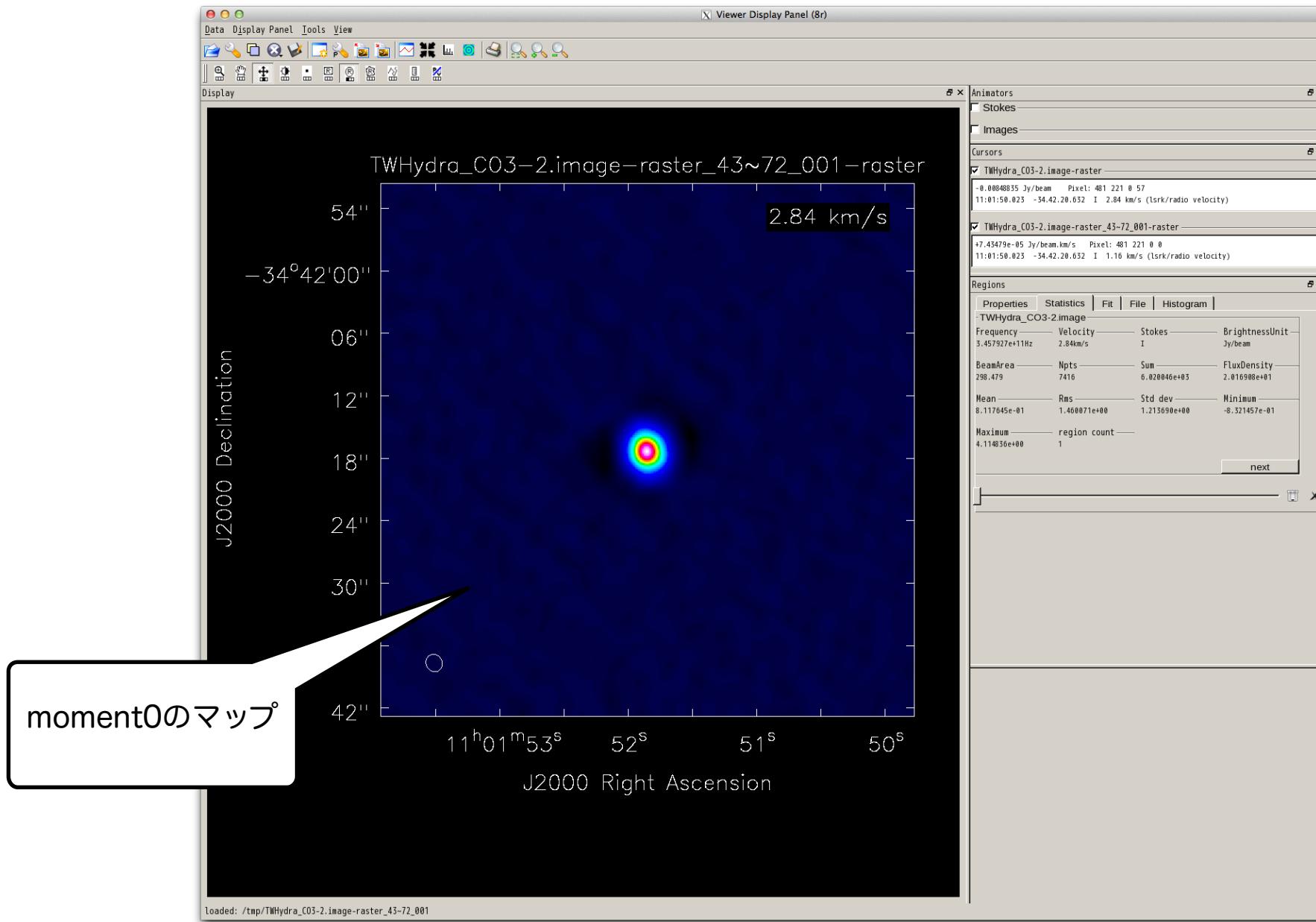


# モーメントマップの作成



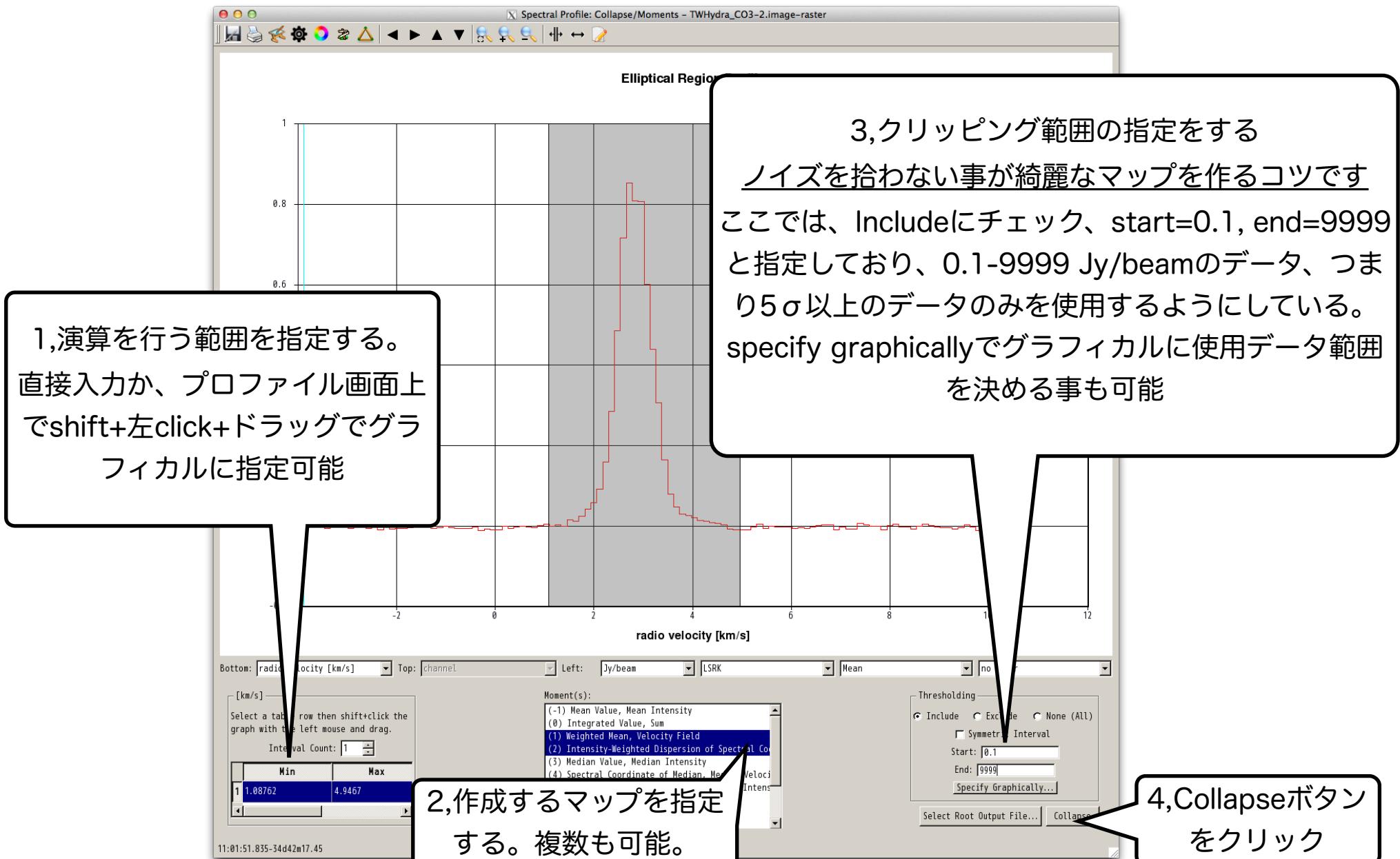
# モーメントマップの作成

- 通常イメージとして出力される。ファイル保存等も可能。



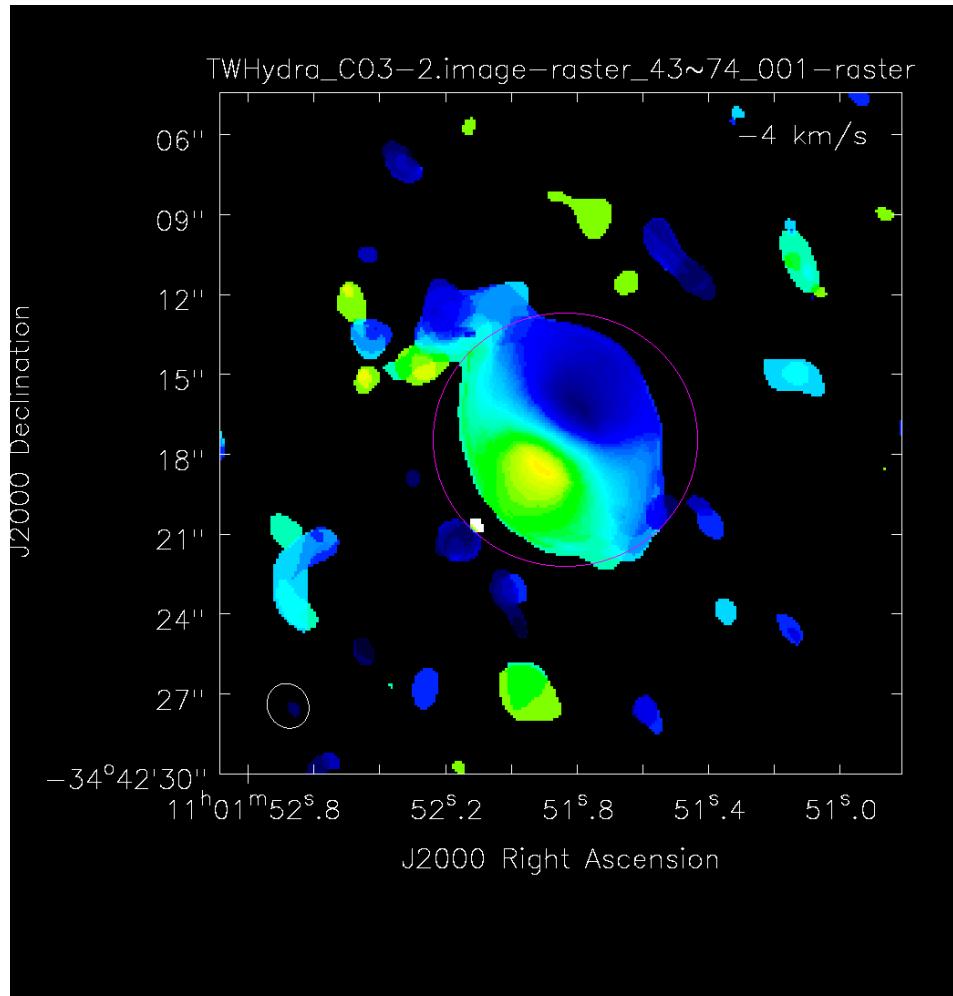
# モーメントマップの作成

- mom1,mom2の場合

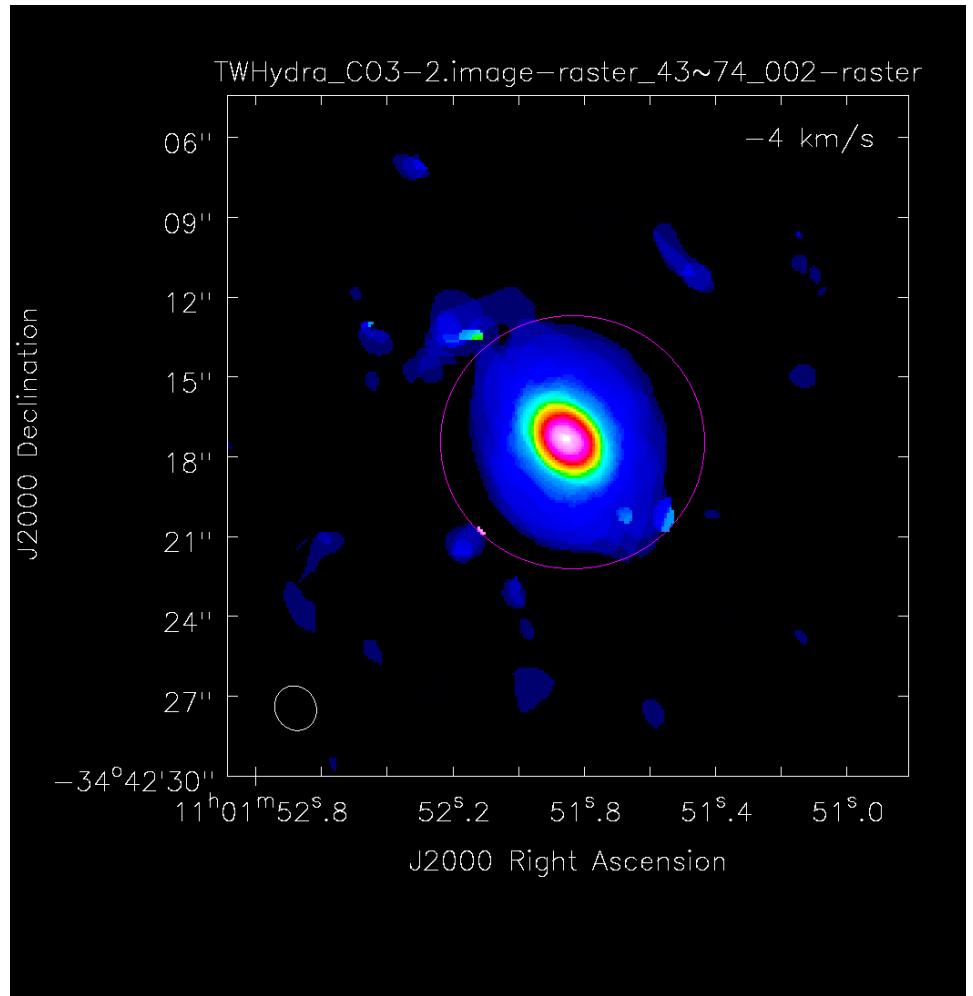


# モーメントマップの作成

moment1



moment2

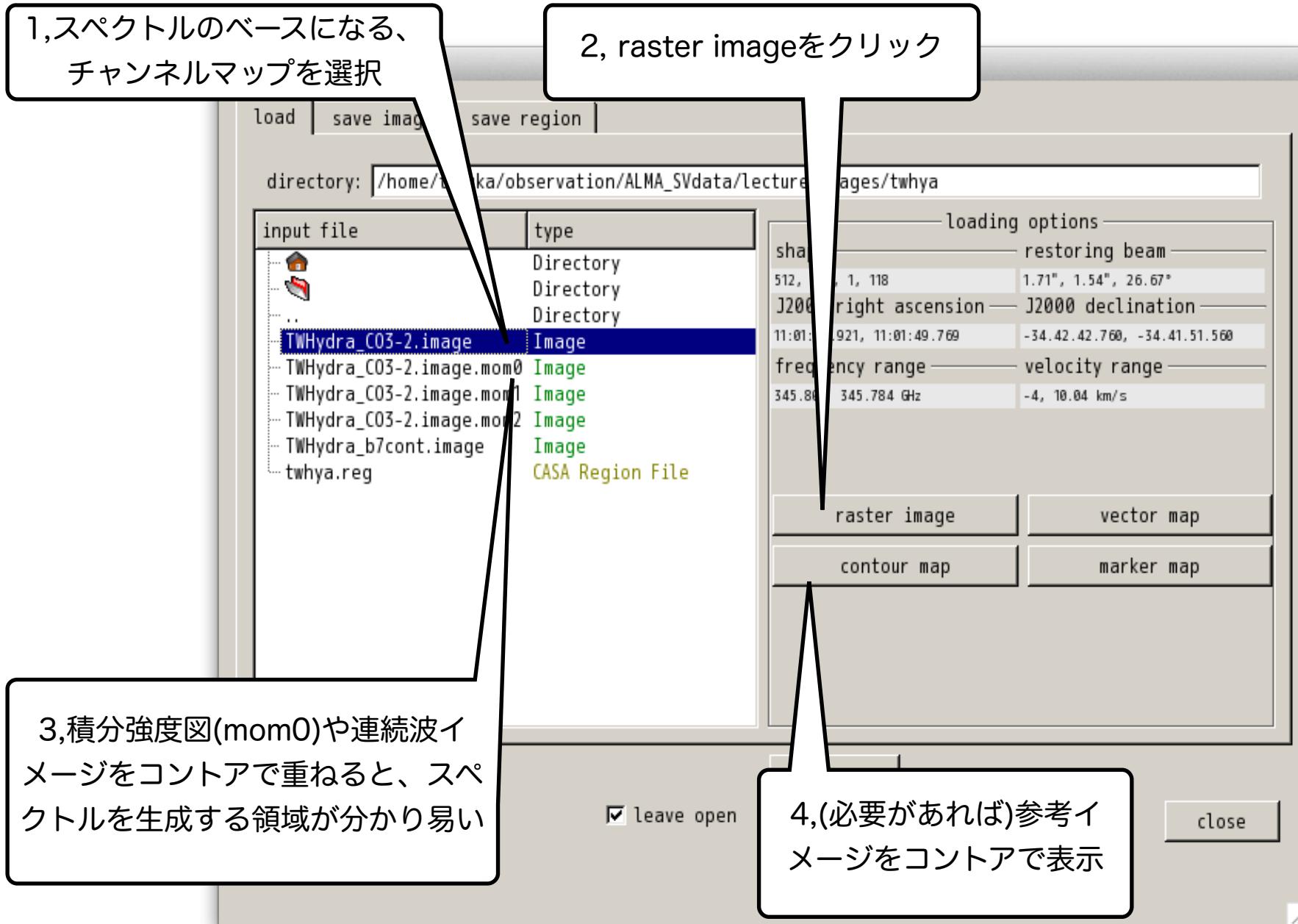


# PV図の作成

# チャネルマップの読み込み

1,スペクトルのベースになる、  
チャンネルマップを選択

2, raster imageをクリック

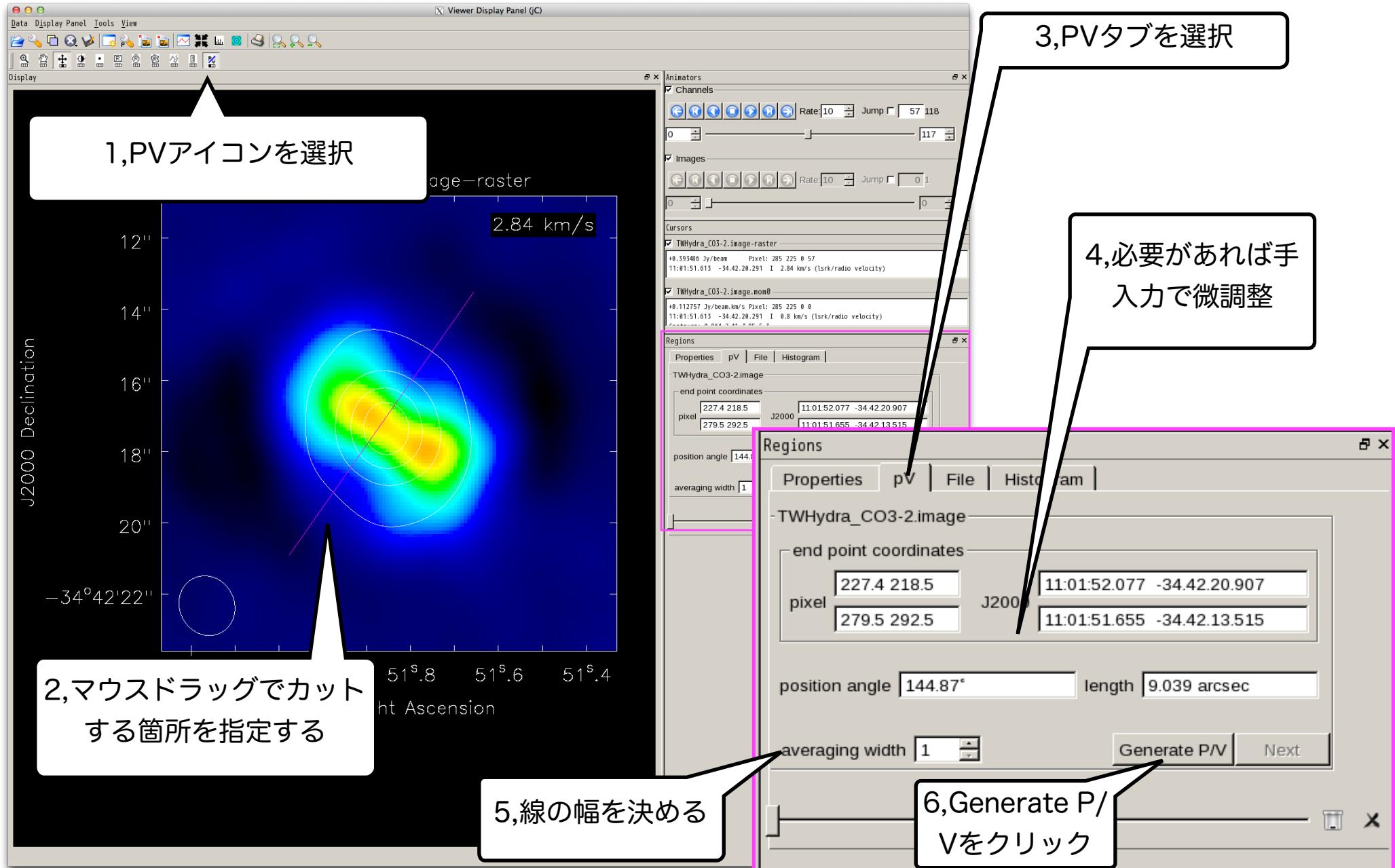


3,積分強度図(mom0)や連続波イメージをコントアで重ねると、スペクトルを生成する領域が分かり易い

4,(必要があれば)参考イメージをコントアで表示

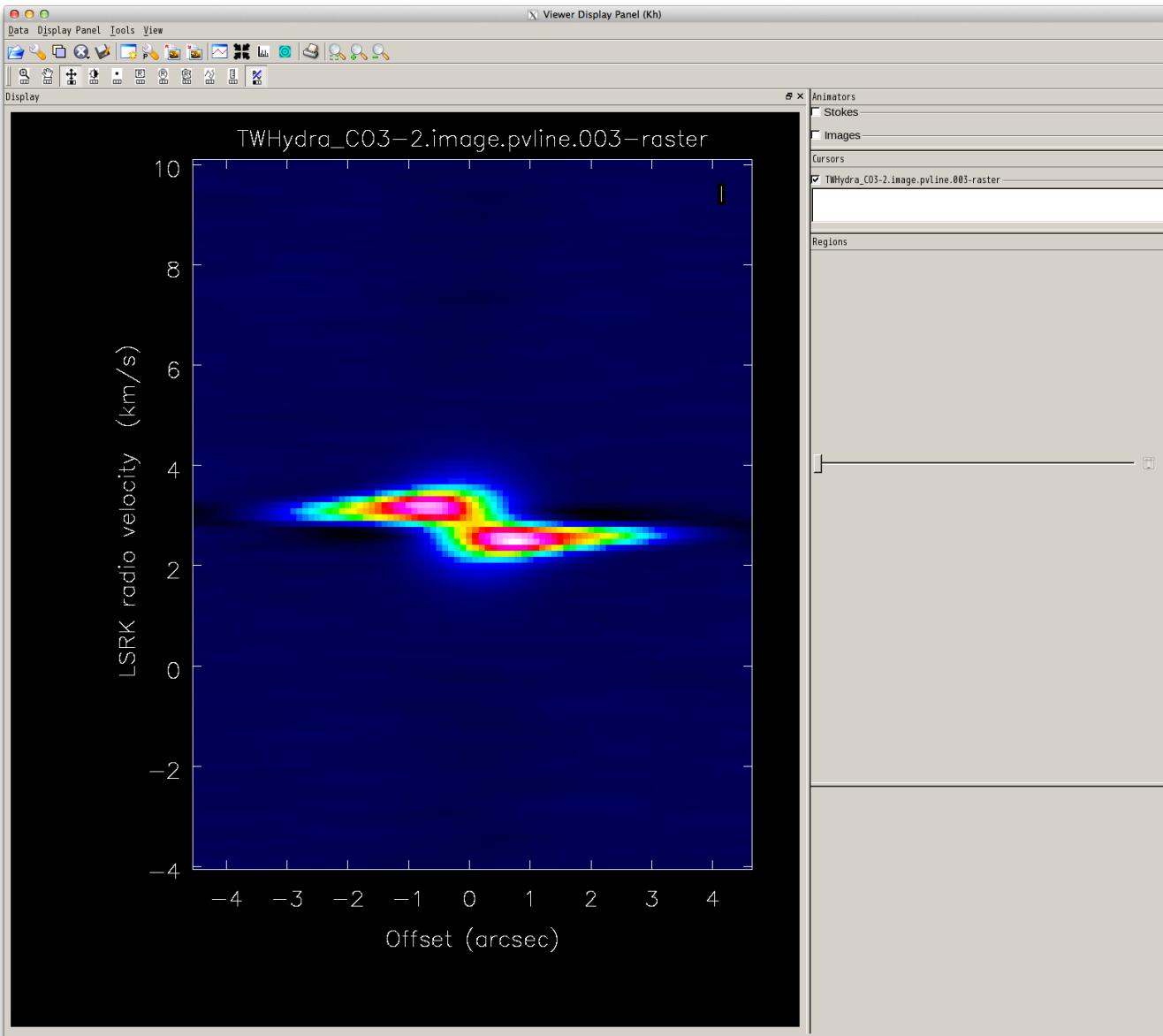
# カットする方向を指定して作成

- ・チャンネルマップ(カラー)+積分強度図(コントア)



# PV図が作成される

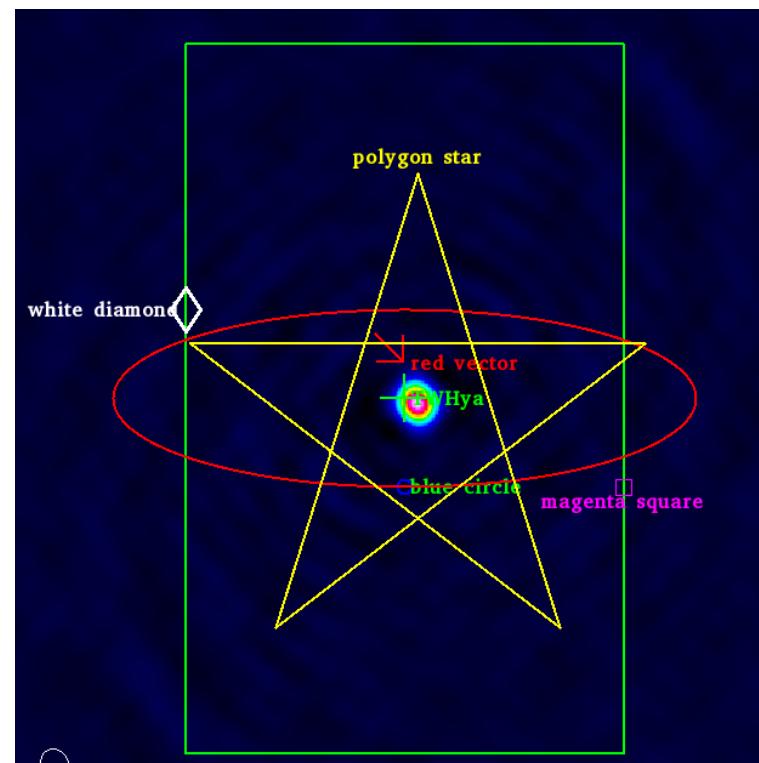
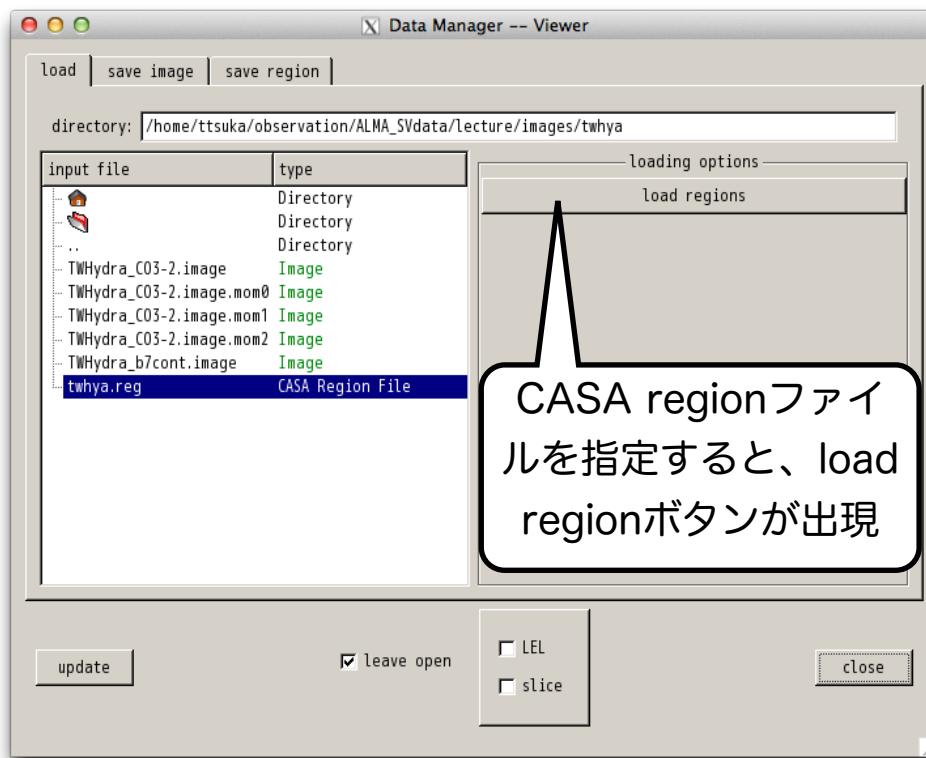
- 通常の画像同様に調整出来ます
- CASAイメージやFITSとして保存可能



# リージョンファイルの使い方

# リージョンファイル

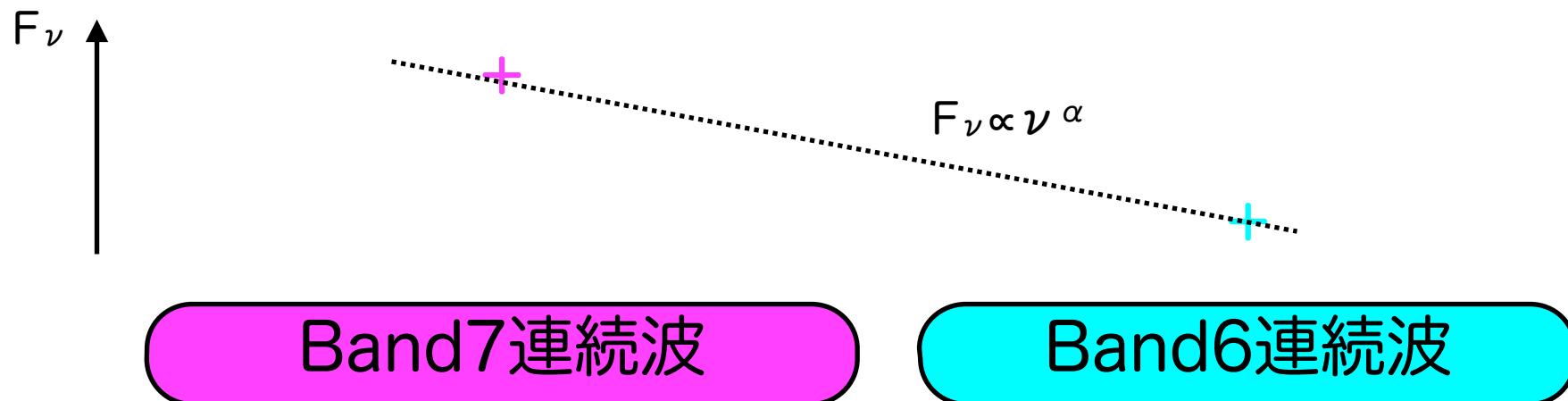
- ・イメージ上の指定した位置にシンボルを置きたい場合に使用
  - ・星の位置や天体名を記載したい、天体の周囲を円やボックスで囲みたい etc.
- ・指定のフォーマットでテキストファイルに記載する。同梱した `twhya.reg` を参照して下さい。
  - ・1行目の頭に`#CRTFv0`と書く事でCASA region formatファイルとして認識される
  - ・[http://casaguides.nrao.edu/index.php?title=CASA\\_Region\\_Format](http://casaguides.nrao.edu/index.php?title=CASA_Region_Format)



# Spectral Indexマップ作成 (advanced)

# スペクトル・インデックスの導出

- 周波数方向に複数のデータが(十分なSNで)取得出来れば、その傾きからスペクトル・インデックスの導出が可能
  - 複数の受信機での観測結果: Band6とBand7など
  - 単一受信機の観測でも、分光計設定で周波数帯域を広く抑えているケース

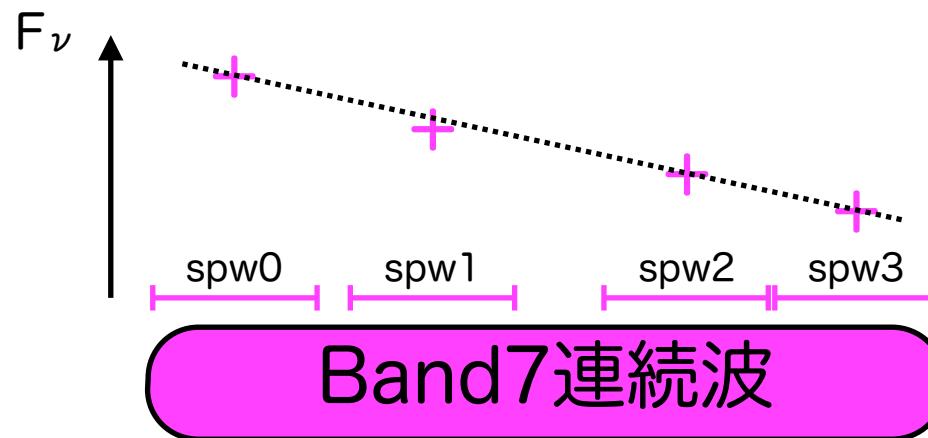


- +帯域が広く取れる
- +多バンドを使えば信頼性上がる
- 各バンド間の絶対強度校正の不定性が残る(5-10%)

HD163296のデータサンプル

# スペクトル・インデックスの導出

- 周波数方向に複数のデータが(十分なSNで)取得出来れば、その傾きからスペクトル・インデックスの導出が可能
  - 複数の受信機での観測結果: Band6とBand7など
  - 単一受信機の観測でも、分光計設定で周波数帯域を広く抑えているケース



- +多バンド間の絶対強度校正の不定性が無い
- +1バンドデータなので取得が容易
- 傾きを出す為の帯域が狭くなる
- 一つの周波数窓の帯域が狭くなり、SNを稼ぎづらい

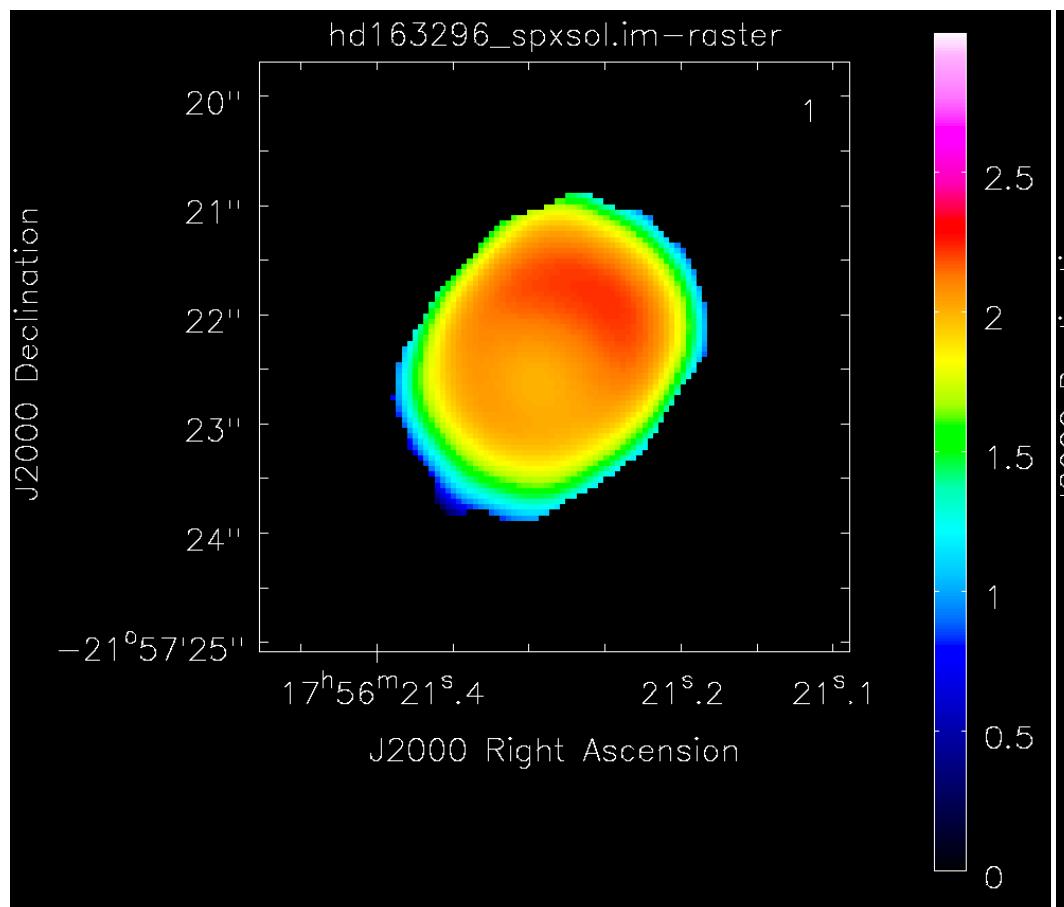
TW Hyaのデータサンプル

# CASAによるスペクトル・インデックス導出

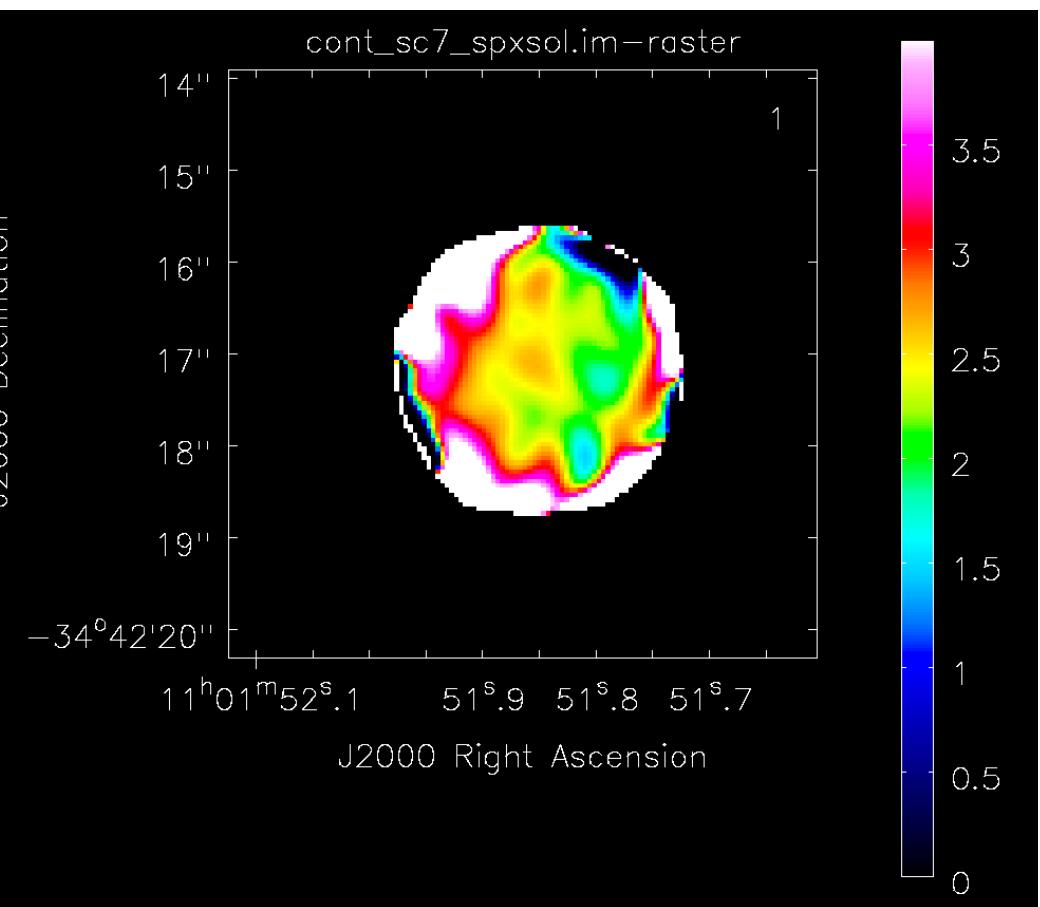
- イメージベースによる導出
- 連続波イメージの作り方に従い、各受信機もしくは各spwに分けたイメージを用意する
  - イメージのサイズやグリッド、合成ビームのサイズ等は合わせておく。イメージング後にリグリッドしても良い。
  - selfcalを行う場合は注意が必要。中心位置をずらさないように補正テーブルを作るか、イメージング後に中心位置を合わせる操作が必要。
- 作成したイメージを用いて、CASAのタスク'spxfit'を実行する
  - 同梱したスクリプトを参照
  - execfile('spindex\_twhya.py'), execfile('spindex\_hd169142.py')
- イメージ上の各ピクセルにおいて $F_\nu = A * \nu^B$ の式でフィットを計算する。出力されるファイル(spxsol)は、2chを持つ3Dキューブとなり、0ch目がA、1ch目がBの値が入る。

# スペクトル・インデックス図の例

HD163296 SVデータ



TW Hya cycle0データ



Band6+7データより

Band7データより